

AD-A086 428

CLARKSON COLL OF TECHNOLOGY POTSDAM N Y

P/B 9/3

LUMPED NONLINEAR SYSTEM ANALYSIS WITH VOLTERRA SERIES, (U)

APR 80 H K THAPAR, B J LEON

F30602-78-C-0102

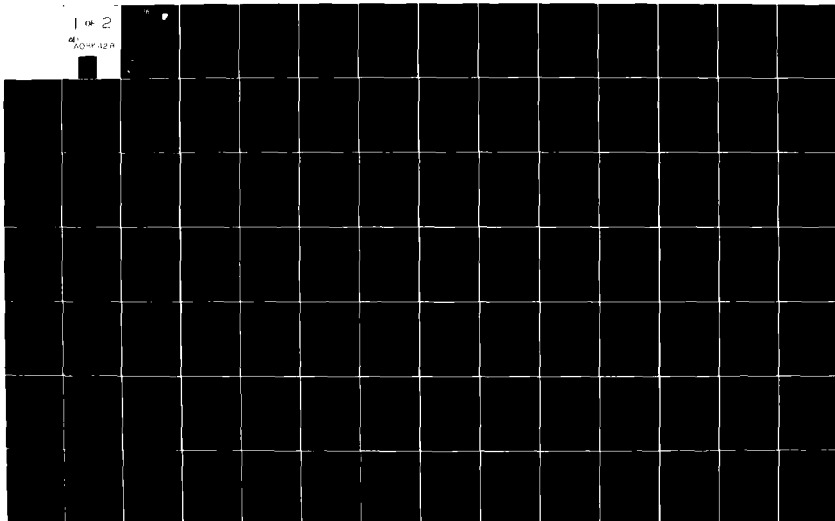
UNCLASSIFIED

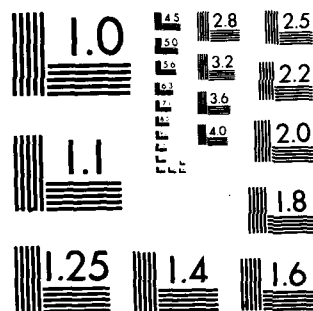
RADC-TR-80-140

NL

1 of 2

AD-A086 428





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

✓  
**LEVEL IV**

12



**RADC-TR-80-140**  
Phase Technical Report  
April 1980

# **LUMPED NONLINEAR SYSTEM ANALYSIS WITH VOLTERRA SERIES**

**Purdue University**

H. K. Thapar  
E. J. Leon

**DTIC**  
**ELECTE**  
**NO 7 80**

ADA086428

LUMPED NONLINEAR SYSTEM ANALYSIS WITH VOLTERRA SERIES

This report has been reviewed by the RADC Public Affairs Office (PAO) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-60-140 has been reviewed and is approved for publication.

APPROVED:

*[Signature]*  
FOR THE DIRECTOR  
RADC-TR-60-140

APPROVED:

*[Signature]*  
FOR THE DIRECTOR  
RADC-TR-60-140

*[Signature]*  
FOR THE DIRECTOR  
RADC-TR-60-140

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
18 1. REPORT NUMBER RADC-TR-80-140 ✓	2. GOVT ACCESSION NO. AD-A086429	3. RECIPIENT'S CATALOG NUMBER 9	
4. TITLE (and Subtitle) LUMPED NONLINEAR SYSTEM ANALYSIS WITH VOLTERRA SERIES,		5. TYPE OF REPORT OR REPORT COVERAGE Phase Report. Nov 78 — Nov 79,	
7. AUTHOR(s) H. K. Thapar B. J. Leon		6. PERFORMING ORG. REPORT NUMBER N/A	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Purdue University West Lafayette IN 47907 Clarkson Coll. of Tech. Potsdam, NY		8. CONTRACT OR GRANT NUMBER(s) F30602-78-C-0102 ✓	
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RBCT) Griffiss AFB NY 13441		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F ✓ 233803PE 1703	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		12. REPORT DATE Apr 80	
		13. NUMBER OF PAGES 175	
		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same			
18. SUPPLEMENTARY NOTES RADC Project Engineer: Jon B. Valente (RBCT)			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Multi-dimensional Transforms Distortion Analysis			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This study investigates the application aspect of the Volterra series method to nonlinear system analysis, with special emphasis on nonlinear circuits. The contents are organized into five parts: (1) important aspects of Volterra series method; (2) characterization of multiple nonlinearity circuits (systems) using Volterra series; (3) response on nonlinear systems from Volterra series system characterization; (4) Computer-aided analysis of mildly nonlinear circuits using Volterra			

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

(Cont'd)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

085000

Jmcc

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Item 20 (Cont'd)

series; and (5) concrete examples from the application of Volterra series method to nonlinear circuit problems.

In presenting the important aspects of the Volterra series method we deal with the applicability and the convergence of the series. Recursive relationships for the upper bound on the error incurred when the Volterra series is truncated are derived for an important class of nonlinear system problems. It is shown how this upper bound can be directly related to the  $l_1$  norm of the linear kernel function and the bound on the input of the linearized system. An example of determining the bound on the input function in terms of the  $l_1$  norm of the linear kernel function to assure the convergence of the series solution is also given.

The characterization of multiple polynomial type nonlinearity circuits (systems) using the Volterra series is done by determining the Volterra Kernels, or their transforms. An approach based on applying multi-dimensional transforms directly to a set of system equations for systematically obtaining the transform domain description of the kernel functions is developed. The case of multiple input circuits is also treated.

In studying the response of nonlinear circuits, relationships between sinusoidal steady state and zero-state transient response and the transform domain description of the kernel functions is given. We make use of the "association of variables" in the transform domain to obtain the zero-state transient response and show how a symbolic analysis on a digital computer will provide both types of solutions - namely, sinusoidal steady-state and zero-state solutions.

Algorithms for adapting the Volterra series method for the computer-aided distortion and zero-state transient analysis of mildly nonlinear circuits are developed next. A symbolic approach is used which not only improves the computational efficiency but also provides a link to the earlier ideas on convergence, etc. An overview of the digital computer program PRANC, which uses the Volterra series method, is also given.

Lastly, we apply PRANC for the determination of intermodulation distortion in a bipolar transistor amplifier.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## PREFACE

This effort was conducted by Purdue University under the sponsorship of the Rome Air Development Center Post-Doctoral Program. Mr. Jon Valente of RADC was the task project engineer and provided overall technical direction and guidance. Prof. B. J. Leon directed this research and the preparation of this report at Purdue University. The authors of the report are B. J. Leon and H. K. Thapar.

The RADC Post-Doctoral Program is a cooperative venture between RADC and some sixty-five universities eligible to participate in the program. Syracuse University, Clarkson College of Technology and the Georgia Institute of Technology act as prime contractor schools with other schools participating via sub-contracts with the prime schools. The U.S. Air Force Academy (Dept. of Electrical Engineering), Air Force Institute of Technology (Dept. of Electrical Engineering) and the Naval Post Graduate School (Dept. of Electrical Engineering) also participate in the program.

The Post-Doctoral Program provides an opportunity for faculty at participating universities to spend up to one year full time on exploratory development and problem-solving efforts with the post-doctorals splitting their time between the customer location and their educational institutions.

Further information about the RADC Post-Doctoral Program can be obtained from Jacob Scherer, RADC/RBC, Griffiss AFB, NY, 13441, telephone AV 587-2543, COMM (315) 330-2543.

This document is the final report for Task 5 of Purdue University's Sub-contract from Clarkson College of Technology. The task was to "Develop and Apply Symbolic Methods to the Volterra Series Approach to Nonlinear Circuit Analysis."

## TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	vii
LIST OF FIGURES . . . . .	viii
LIST OF ABBREVIATIONS . . . . .	ix
CHAPTER 1 - INTRODUCTION. . . . .	1
1-1. Introduction . . . . .	1
1-2. Nonlinear System Analysis Methods. . . . .	2
1-3. Objectives of the Investigation. . . . .	6
1-4. Organization of the Report . . . . .	9
CHAPTER 2 - VOLTERRA FUNCTIONAL SERIES. . . . .	12
2-1. Introduction . . . . .	12
2-2. Volterra Functional Series . . . . .	13
2-3. Derivation of the Volterra Series: An Example . . . . .	18
2-3. Convergence properties of Volterra Series. . . . .	24
CHAPTER 3 - MULTIPLE-NODE, MULTIPLE-NONLINEARITY CIRCUIT ANALYSIS . . . . .	47
3-1. Introduction . . . . .	47
3-2. Multi-dimensional Transforms . . . . .	49
3-3. A Nonlinear Differential Equation. . . . .	56
3-4. Multiple-node, Multiple-nonlinearity Circuit Analysis. . . . .	64
3-5. Multiple Input Circuit Analysis. . . . .	78
3-6. An Example . . . . .	83
CHAPTER 4 - STEADY-STATE AND TRANSIENT ANALYSIS . . . . .	90
4-1. Introduction . . . . .	90
4-2. Sinusoidal Steady-State Analysis . . . . .	90
4-3. Transient Analysis Using Volterra Series . . . . .	98
CHAPTER 5 - COMPUTER-AIDED ANALYSIS USING VOLTERRA SERIES . . . . .	112
5-1. Introduction . . . . .	112
5-2. Why a Symbolic Approach. . . . .	114
5-3. Symbolic Analysis Method . . . . .	117
5-4. Spectrum and Distortion Analysis Algorithm . . . . .	124
5-5. Time-Domain Analysis Algorithm . . . . .	128
5-6. Program PRANC. . . . .	132



CHAPTER 6 - NONLINEAR CIRCUIT ANALYSIS EXAMPLES . . . . .	138
6-1. Introduction . . . . .	138
6-2. Spectrum and Distortion Analysis Example . . . . .	139
6-3. Time-domain Analysis Example . . . . .	147
6-4. PRANC vs. NCAP . . . . .	149
CHAPTER 7 - CONCLUDING REMARKS. . . . .	151
7-1. Summary. . . . .	151
7-2. Further Research . . . . .	152
REFERENCES. . . . .	154
APPENDIX. . . . .	160

Accession For	
NTIS G.M&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Identification	
By	
Distribution/	
Availability Codes	
Dist	Available/or special
<i>A</i>	

## LIST OF TABLES

TABLE	Page
3-1. Nonlinear Current Sources in multiple-node, multiple-nonlinearity circuit analysis . . . . .	73,74
4-1. List of Associated Transforms. . . . .	102
6-1. Transistor Parameters for the Circuit of Fig. 6-1. . . . .	139
6-2. Coefficients of the polynomial type nonlinearities . . . . .	143
6-3. Summary of Truncation Error . . . . .	149

# LIST OF FIGURES

Figure	Page
2-1. A Time-invariant Nonlinear System. . . . .	13
2-2. Volterra Series Equivalent System Representation . . . . .	17
2-3. A System with "non-PS" kernel. . . . .	35
2-4. Roots of the Nonlinear Algebraic Equation. . . . .	45
3-1. Step in Nonlinear Circuit Analysis Using Volterra Series . . . . .	68,69
3-2. Determination of Volterra Transfer Functions . . . . .	77
3-3. Multiple Input Nonlinear Circuit Analysis. . . . .	80,81
3-4. A Nonlinear Circuit Example. . . . .	84
5-1. Determination of $Z(s)$ for the p-port network using state equations. . . . .	120
5-2. Algorithm for Inverting $Y(s)$ symbolically. . . . .	123
5-3. Algorithm for Spectrum and Distortion Analysis . . . . .	126
5-4. Algorithm for Computing time-domain response using Volterra series. . . . .	131
6-1. A Transistor Amplifier Circuit . . . . .	140
6-2. Transistor equivalent model. . . . .	142
6-3. First-order Transfer function magnitude characteristics. . . . .	144
6-4. Second-order Transfer function magnitude characteristics . . . . .	144
6-5. Harmonic Distortion Variation with amplitude . . . . .	146
6-6. A Single Nonlinearity Circuit Example. . . . .	147

#### LIST OF ABBREVIATIONS

FM - Frequency Modulation

PS - Product Structure

PFE - Partial Fraction Expansion

## CHAPTER 1

### INTRODUCTION

#### 1-1. Introduction

The superposition property shared by all linear systems makes it possible to develop unified methods of analysis and synthesis of these systems. By contrast, nonlinear systems do not share this property and as such no single method exists at this time which can be applied to the analysis and synthesis of all such systems.

The area of nonlinear systems analysis, however, remains an important area of research. Most devices which are linear in a certain region of operation may reveal a nonlinear behavior when operated in a different region. Devices such as diodes, transistors, tunnel diodes, etc., which are ubiquitous to all electronic circuits, are inherently nonlinear (except in a very small region of operation). The analysis of circuits containing such devices is an important problem.

In the analysis of nonlinear systems, two alternatives are available: The first possibility is to take a problem-dependent analysis approach; that is, take a specific nonlinear system and perform a detailed analysis on it, and, in the process, develop some efficient method of analysis. In general, this "new" method may not be useful for other nonlinear system problems. The second alternative is to develop and apply an analysis method which is applicable to a broad class of nonlinear systems, and is not oriented towards a specific system.

In the present day, when computer-aided analysis pervades many areas of science and engineering, the computer has become an important arbiter in the success and popularity of an analysis method. Even a fairly general analysis method may remain in oblivion if it is found - or thought - unadaptable to computer-aided analysis.

The Volterra functional series [1,2] for nonlinear system analysis may be regarded as one such method: it is applicable to a large class of nonlinear system problems, but is not widely used in engineering calculations. Although much theoretical work [3-6,12-14,15-24] has been done on this method over the past two decades, the applications of this method have been limited in their scope. Most authors of these publications have ignored the computational aspect of this method, and those who have considered it have deemed this method as cumbersome or unadaptable to computer-aided analysis. In this report we will use the Volterra series as a basis for computer-aided circuit analysis of nonlinear circuits, and, in the process, introduce some interesting aspects of the method.

An overview of some nonlinear system analysis methods is given in section 1-2, with special emphasis on the Volterra series approach. Section 1-3 presents the objectives of this investigation. Finally, a glimpse of the results of this research effort, along with the organization of this report, is presented in section 1-4.

#### 1-2. Nonlinear System Analysis Methods.

It is indeed rare to find a nonlinear system analysis problem which has a closed-form solution. One must then resort to using approximation methods for gaining insight into the system behavior. The broad ca-

category of approximation methods can be roughly broken into two groups: 1) analytical methods, which often yield qualitative and quantitative information about the system, and 2) numerical methods, which give quantitative information about the system. The well-known methods belonging to the latter group are: 1) polynomial approximation to the solution function, leading to the Numerical Integration methods; and 2) Taylor series approximation of the solution, leading to the Runge-Kutte Type methods. Our survey on the approximation methods will deal with the analytical methods only. We consider some of them in the following sub-sections.

1-2.1 Iteration Method. A method for solving nonlinear differential or integral equations is based on the process of successive iteration, called the iteration method. This process can be performed in a number of ways, but the basic procedure is the same. The nonlinear equation is first solved by neglecting certain terms - generally the nonlinear terms - in the equation. The resulting solution is then re-substituted in the system equation without neglecting any terms. This process is repeated and, under well defined conditions [15,18,35], each resulting solution is a better approximation to the actual solution.

When nonlinear differential/integral equations with polynomial type nonlinear terms are solved via the iteration process, the implementation involves the actual squaring, cubing, etc., operations. When sinusoidal steady state response is obtained using this method, the approximation to the entire output spectrum is obtained in one step, in contrast to some other methods, which give new distortion terms with each step.

1-2.2 Perturbation Method. Like the iteration method, the perturbation

method is applied in a wide variety of ways. This method is generally applied to nonlinear differential equations, in which a small parameter is associated with the nonlinear terms, by introducing a dummy variable to help dissolve the nonlinear differential equation into a sequence of linear differential equations with nonlinear terms which can be solved in a bootstrapping manner. The linearized system is first solved. The resulting solution is used to solve the next system equations involving the quadratic terms. This is followed by solving the linear differential equations with cubic terms, and so on. The solutions obtained at each step are added to yield the approximation to the actual solution, thus giving a series type approximation.

The description of the procedure above is the basic perturbation method. However, in some nonlinear problems this approach, when applied as above, leads to serious convergence problems. The method is then modified to get rid of the so-called "secular terms," which grow up indefinitely as  $t \rightarrow \infty$  [17,18].

1-2.3 Volterra Series Method. The Volterra series method is a type of functional series which relates the system input,  $x(t)$ , to the system output,  $y(t)$ , as:

$$y(t) = \sum_{n=1}^{\infty} \int \dots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i \quad (1-1)$$

where  $h_n$  is the  $n$ -th order kernel function, whose  $n$ -dimensional transform is called the  $n$ -th order transfer function. Clearly, the system characterization, using Volterra series, is done by determining these kernel functions, or their transforms. The solution obtained is



of a series type.

Wiener [2] first applied this method to calculate the response of a nonlinear system with memory to a Gaussian noise input. Much of theoretical work on this method was done in the late 50's and early 60's. Brilliant [4] studied the convergence of the series for bounded inputs. George [5] devised a "system of algebra" for combining Volterra systems and developed the "association of variables" technique for directly going from a multi-dimensional transform description to the one-dimensional transform description. Zames [3] studied Volterra systems when placed in a feedback loop. Bedrosian and Rice [12] and Rudko and Wiener [37] give a set of formulae for Volterra systems driven by random inputs. This set of formulae have been applied to Gaussian random inputs only.

The approach has also been applied to specific problems. Van Trees [9] characterized the phase-locked loop with a nonlinearity. Bedrosian and Rice [12] and Baranyi [38] used the method to calculate and compensate the distortion incurred in an FM signal, respectively. Narayanan [7] and Gopal, et. al. [36] calculated the intermodulation distortion in transistor amplifiers. The modeling of communication receivers using Volterra series was done in [10]. Ewen [11] studied the identification of Volterra systems and Naditch [13] applied the approach for high frequency calculations. The relationship between Volterra series and Picard iteration, for a class of nonlinear system problems, was developed by Leon and Shaefer [19].

### 1-3. Objectives of the Investigation.

In the analysis of nonlinear systems, two main classes of solutions are generally sought: 1) transient, and 2) steady state. The basic goal of this investigation is to obtain these solutions for nonlinear circuits via the Volterra series method. Before proceeding with the main subject of this sub-section, we briefly provide the motivation for using Volterra series.

The most commonly used present-day approach for analyzing nonlinear systems is numerical integration. The nonlinear differential equations are integrated from some initial time,  $t_0$ , to some final time,  $t_f$ . When the sinusoidal steady state response is sought, the value of  $t_f$  chosen is usually large to insure that all transients have decayed. A fast Fourier transform then yields the frequency components of the output response. A more efficient method for obtaining the sinusoidal steady-state response is to pose the analysis problem as a two-point boundary value problem [31,32] and then apply an iterative two step algorithm. This approach, however, allows only single frequency inputs.

The problems involved in the numerical integration method are well known [29]. These problems notwithstanding, there are other inefficiencies. When one is solely interested in the sinusoidal steady state response, the computation expended in reaching close to  $t_f$  is a waste. This inefficiency grows as poles of the linearized system get close to the  $j$ -axis, as is often the case in many quasi-linear communication circuits.

Other methods such as the harmonic balance or the describing function method are seldom used, simply because the assumption behind these

methods render them undependable. The Picard iteration method can also be used for nonlinear system analysis. This method also has limitations when used for computer-aided analysis, particularly when multi-tone inputs are present. The basic perturbation method and Volterra series are alike when applied to nonlinear circuit analysis problems.

We now return to the main subject of this sub-section, namely, the goals of this investigation.

As mentioned in the previous sub-section, the nonlinear system is completely characterized by the Volterra kernel functions, or their multi-dimensional transforms, referred to as Volterra transfer functions. In analyzing a nonlinear circuit, the first problem therefore is to characterize it.

Once the kernel functions or the transfer functions are known, we can determine the response of the circuit. However, since the Volterra series is an infinite series, the error incurred as a result of truncation at a finite term must be investigated. This leads to the problem of convergence. We look into how the series converges for a class of lumped nonlinear circuits.

Next we investigate the implementation of the Volterra series method on a digital computer. In doing so, we first develop the basic algorithm for adapting this method for computer-aided analysis and then actually implement it for obtaining the steady-state response of circuits with multiple nonlinearity, multiple multi-tone input sources. The determination of the transient response from the Volterra transfer function for Laplace transformable inputs is also investigated.

In summary, the goals of this investigation can be categorically stated as:

- 1) Devise a systematic way of deriving the Volterra kernel functions, or their transform domain description.
- 2) Determine a method of estimating the error incurred as a result of truncating the series solution at a finite term.
- 3) Obtain an algorithm for adapting the Volterra series method for analyzing multiple nonlinearity circuits.
- 4) Implement the algorithm of (3) as a computer program.

Before concluding this subsection, we make the following comment: Many papers and theses on the theoretical aspects of Volterra series have appeared in the literature, most of which have been included in the References. After studying these references, one is still left with the most practical questions: How do we apply this method to the real world nonlinear circuit problems? How do we use this approach for analyzing and computing the behavior of a nonlinear system problem at hand?

The motivation behind this research was to seek answers to the above fundamental questions and also remove the misconception that the Volterra series method is cumbersome for computer analysis. We feel that this method could be employed in a wider variety of engineering calculations than is presently the case. The contents of this report are therefore centered around the practical aspects concerning Volterra series.

#### 1-4. Organization of the Report

In addition to this introductory chapter, this report comprises six chapters.

Chapter 2, entitled "Volterra Series", discusses the analysis method (1-2.3) which forms the basis of this investigation. Commonly used terminology and the salient features associated with the series are presented. The applicability along with the convergence of the Volterra series is discussed. In treating the topic of convergence, we concern ourselves not with abstract nonlinear equations, but with a concrete class of time-invariant nonlinear system equations with polynomial type nonlinearities. Recursive relationships, which provide an estimate of the error due to truncating the series (1-1), for this class of nonlinear differential equations are developed. Knowing the  $l_1$  norm of the linear kernel function, one can estimate the number of kernel functions that needs be derived to meet a prescribed error criterion. When the solution is sought during a finite time interval, the conditions for which the Volterra series converges are given in [21,22]. However, for the interval  $[-\infty, \infty]$ , convergence of the series has not been proven. We present an example, based on [23] and the new recursive relationships, to show how the bound on the input, in terms of the  $l_1$  norm of the linear kernel function, for which the series in this interval converges can be derived.

In Chapter 3, we deal with the topic of multiple nonlinearity circuit analysis. In dealing with this topic we develop a systematic method of obtaining the transform domain description of the Volterra kernel functions for a large class of nonlinear systems. Our approach

relies on the application of multi-dimensional transforms, in contrast to the "harmonic input" method used by other authors [7,12-14]. This new approach, besides being more intuitive, avoids the morass of algebra involved in the other approaches. The determination of Volterra transfer functions for multiple nonlinearity circuits with single inputs is described. The extension to multiple inputs is also discussed briefly.

Chapter 4 deals with the determination of the sinusoidal steady-state and zero-state transient responses from the Volterra transfer functions. The case of multi-tone inputs is considered, and the determination of the responses for calculating the distortion indices at the various and many frequencies is discussed. The fact that convolution in the time-domain becomes multiplication in the transform domain, along with the "association of variables" technique, is exploited to show how the zero-state transient response can be determined for inputs which are Laplace transformable and factorable. Nonlinear lumped systems with such inputs obviate the need for any numerical integration. With the complete pole-residue information of the linearized system, it is shown that one can determine the time-domain response for such inputs.

The ideas gathered in Chapters 2, 3, and 4 are used to develop algorithms for most "efficiently" implementing the Volterra series method for nonlinear circuit analysis on a digital computer. This is the topic of Chapter 5. The fundamental idea used in this development is the use of semi-symbolic [29] analysis of the linearized circuit. Through this approach, many advantages are accrued: 1) repeated inversion of a "large" matrix is avoided; 2) the pole-residue information obtained can

be used for zero-state transient analysis; 3) the pole-residue information is directly related to the  $L_1$  norm of the linear kernel function, which, in turn, is related to the truncation error (Chapter 2); 4) the functional description of the higher order transfer functions can be obtained from the functional description of the linear functions (Chapter 3). Thus, Chapter 5, in a sense, lends unity to this whole investigation.

Chapter 6 gives an example from the use of PRANC. Various ramifications from the results obtained are discussed.

Chapter 7 is reserved for concluding remarks and some topics of further research.

CHAPTER 2  
VOLTERRA FUNCTIONAL SERIES

2-1. Introduction

In many of the problems encountered in nonlinear systems analysis, it is difficult - if not impossible - to find explicit closed-form expressions for the solution. Approximation for the solution must then be sought. Several methods [16-18] which yield approximate solution are available, and were discussed in the previous section.

The intention behind using the Volterra functional series is fundamentally to evaluate solutions which cannot be obtained in closed form, and therefore the solution obtained from this approach will involve an approximation. Whenever approximation methods are used, the question of convergence is an important consideration. In this section we present the general convergence properties of the Volterra series method. In dealing with this topic we rely on the results of [22]. These results are extended to an important class of nonlinear systems by establishing the bounds on the output in terms of the bound on the input and the  $l_1$  norm of the impulse response of the linearized system. We believe that these results are of more practical value to a nonlinear system analyst and designer.

Other theoretical considerations, such as the applicability of the method and input-output uniqueness are also briefly discussed. This discussion relies heavily on the works of George [5], Brilliant [4].

In section 2-2 we present the Volterra functional series representation of nonlinear systems along with the applicability of the method. In discussing the applicability we present a sufficient condition [4]



for a nonlinear system to be representable by the functional series method.

Section 2-3 is used to illustrate the application of the Volterra series method for obtaining the approximate solution of a general nonlinear time-invariant differential equation.

Section 2-4 deals with the convergence properties of the Volterra functional series. Both the single nonlinearity case and the multiple nonlinearity case are discussed in this sub-section. A brief exposition on the stability of solutions is also presented. The stability considered here is that the response,  $y(t)$ , of the system remains bounded; that is,  $y(t) < \infty$ , for all  $t \geq t_0$ .

## 2-2. Volterra Functional Series

Consider the time-invariant nonlinear system of Figure 2-1, with an input,  $x(t)$ , and an output,  $y(t)$ . The nonlinear system acts as an operator that maps the function  $x(t)$  into a function  $y(t)$ ; that is

$$y(t) = S[x(t)] \quad (2-1)$$

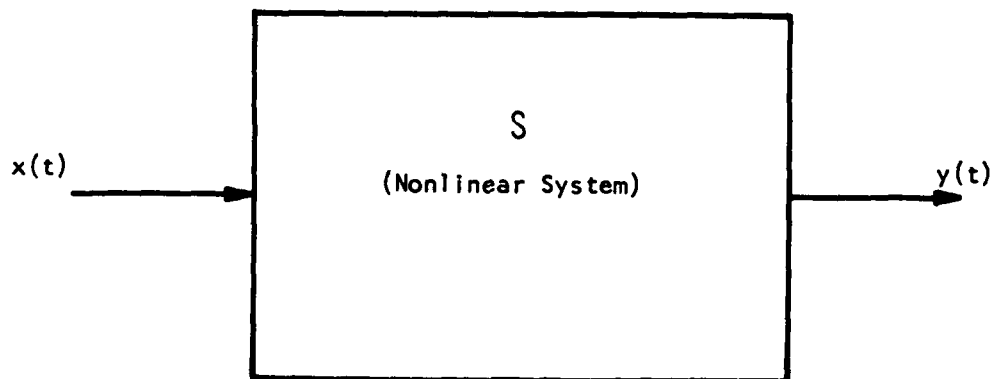


Fig. 2-1. A time-invariant nonlinear system

The basic idea behind the Volterra functional series is to approximate  $S$  by a series of functionals. This idea is analogous to the case of functions, where any function, under some well-defined conditions, can be approximated by a series of functions, say, for example, polynomials. Furthermore, for polynomials, the Weierstrass theorem shows that any continuous function can be approximated arbitrarily closely in the limit in a region by a sequence of polynomials [33]. Similarly, it has been shown\* that for a continuous functional  $S[x(t); t_0 < t < t_1]$ , there exists a sequence of functionals which approximate  $S$  arbitrarily closely in the limit. While the existence of these sequence of functionals is proved, their method of determination is not given.\*\* The Volterra functional series is therefore a type of functional series used for approximating  $S$ . It relates the input,  $x(t)$ , to the output,  $y(t)$ , as follows:

$$y(t) = \sum_{n=1}^{\infty} \int \dots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{p=1}^n x(t-\tau_p) d\tau_p \quad (2-2)$$

where  $h_n(\tau_1, \dots, \tau_n)$  is known as the time-domain Volterra kernel of order  $n$ . It should be clear that the nonlinear system characterization is done through the determination of these kernel functions; for, once we know them, the output function can be determined from eqn. (2-2) for a known input function. For causal kernel functions, the limits of integration are between 0 and  $\infty$ , i.e.  $t_0$  is taken to be zero.

---

\*Van Trees [8] claims that this was done by Frechet. For original reference see [8].

\*\*This again is analogous to the case of functions: the existence of the sequence of approximating functions is proved by Weierstrass, but a method of finding these functions is not given. We then resort to using orthogonal polynomials, etc. as basis functions.

Two well-known cases are readily derived from eqn. 2-2: First, when the set of kernels  $\{h_2, h_3, \dots\}$  are zero, eqn. (2-2) reduces to the well-known convolution integral of a linear system; and, two, when the kernel  $h_n$  is given by  $a_n \prod_{p=1}^n \delta(\tau_p)$ , then eqn. (2-2) represents the output of a nonlinear, memoryless system.

Before proceeding with the discussion of the applicability of Volterra functional series, we point out some important features and definitions associated with the series:

(a) The term order with respect to an output component is often used. Order in this context is defined as the number of times the input is multiplied with itself in the convolution integrals appearing in eqn. 2-2. Due to this definition of order, it is often more convenient to write the series as:

$$y(t) = \sum_{n=1}^{\infty} y_n(t) \quad (2-3)$$

where

$$y_n(t) = \int \dots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{p=1}^n x(t-\tau_p) d\tau_p \quad (2-4)$$

Equation (2-3) then represents the output,  $y(t)$ , as a summation of the outputs of order 1, 2, 3, ....

(b) As shown in eqn. (2-4), the  $n$ th order output functional is equal to the convolution of the kernel  $h_n$  with the  $n$ -fold product of the system input values. Thus, if the input is an impulse function, then  $h_n$  can be viewed as the nonlinear impulse response function of order  $n$ ,

whose  $n$  dimensional transform can then be called the nonlinear transfer function of order  $n$  [10,14].

(c) The Volterra series can be used to characterize nonlinear systems in which the present output depends on the present and past values of the input function. Such systems are referred to as "systems with memory".

(d) The kernel functions themselves are not necessarily independent of each other, but they are independent of the system input. Thus, a parallel system realization of the form shown in Fig. 2-2 describes the nonlinear system being studied. For obtaining an  $n$ th order response, it follows that the first  $n$  kernel functions, or their transform domain description, must be determined.

(e) Some important advantages of the Volterra series method are that it places the input-output relation in explicit form and allows us to think of the system in terms of functional blocks, as shown in Fig. 2-2.

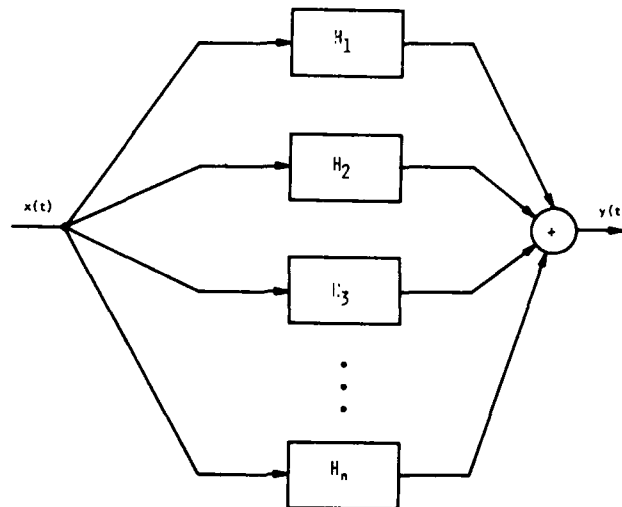


Fig. 2-2. Volterra series equivalent system representation.

This often allows us to deduce qualitative properties about the system which may not be clear from the differential/integral equations used to describe the system.

We now proceed with an important aspect of the Volterra series method: its applicability. Much of the recent work on this topic was done by Brilliant [4], who introduced a new topological space to define the concept of a continuous system mathematically and showed that such systems could be approximated by the functional representation.

Once again consider the nonlinear system of Fig. 2-1. Let  $x_1(t)$  and  $x_2(t)$  be the inputs to the system which produce  $y_1(t)$  and  $y_2(t)$  as outputs, respectively. We define the distance between the input functions and the output functions as follows:

$$d_1(x_1, x_2) = \left| \int_{t-t_1}^t [x_1(\tau) - x_2(\tau)] d\tau \right|, \quad t_1 > 0 \quad (2-5)$$

and

$$d_2(y_1, y_2) = |y_1(t) - y_2(t)| \quad (2-6)$$

where  $y_1(t) = S[x_1(t)]$  and  $y_2(t) = S[x_2(t)]$ . Intuitively, we can say that  $S$  is a continuous system if  $d_2$  is small when  $d_1$  is small. Brilliant [4] gave the following mathematical definition of a continuous system: For a time-invariant system  $S$  with bounded inputs  $x_1(t)$  and  $x_2(t)$ ,  $S$  is continuous if for any  $\epsilon > 0$ , there exists a  $\delta > 0$ ,  $T > 0$  ( $T$  sufficiently large,  $\delta$  sufficiently small) such that if  $d_1 < \delta$  for  $0 \leq t_1 \leq T$ , then  $d_2 < \epsilon$ . It has been shown [4] that if  $S$  is continuous, according to the above definition, then for any  $\epsilon > 0$  there is a polynomial system  $S_\epsilon$  such that, for a bounded input  $x(t)$ ,  $|S[x(t)] - S_\epsilon[x(t)]| < \epsilon$ .

It should be noted that the distance functions defined in eqns. (2-5) and (2-6) are not unique, and that other functions could have been chosen. The definition of continuity stated above is just one definition and, therefore, the aforementioned condition for  $S$  to admit a functional series representation is only a sufficient condition, and not a necessary one.

The conditions on the system  $S$  that admits a Volterra functional series description can be qualitatively summarized as follows:

- (a) Output is a single valued function of the input.
- (b)  $S$  is time-invariant.\*

---

\*The Volterra functional series has been applied to time-varying systems also. See, for example, [39].

- (c) Small perturbations in the input do not produce abrupt changes in the output.

### 2-3. Derivation of the Volterra Series: An Example

The basic problem in applying the Volterra series method to non-linear systems is the determination of the kernel functions. In this section we present an approach for deriving these kernel functions. A direct and more algorithmic approach, which also shows the similarity between the Volterra series method and the perturbation method [17], is given in section 3.

We consider a system with a single nonlinear element and the following differential equation which provides the input-output relationship:

$$L_1[y(t)] + \epsilon L_2\left[\sum_{n=2}^N a_n y^n(t)\right] = x(t) \quad (2-7)$$

where  $L_1$  and  $L_2$  are linear operators, defined as follows:

$$L_1[\cdot] = \frac{d^m}{dt^m} [\cdot] + b_{m-1} \frac{d^{m-1}}{dt^{m-1}} [\cdot] + \dots + b_0 [\cdot] \quad (2-8)$$

and

$$L_2[\cdot] = \epsilon_1 \frac{d}{dt} [\cdot] + \epsilon_2 \int_{-\infty}^t [\cdot] dt + \epsilon_3 [\cdot] \quad (2-9)$$

where  $\epsilon_i$  and  $b_i$  are constants. Notice that if  $\epsilon_1 = \epsilon_2 = 0$ , we have a memoryless nonlinearity such as a resistor; when  $\epsilon_1$  or  $\epsilon_2$  are non-zero, we have a memory-type nonlinearity such as a capacitor or an inductor.

Thus, eqn. (2-7) describes a large class of nonlinear systems. We now proceed to determine the various kernel functions.

If  $\epsilon = 0$ , eqn. (2-7) is the linear differential equation:

$$L_1[y(t)] = x(t) \quad (2-10)$$

Denoting the impulse response of this differential equation by  $h(t)$ , then eqn. (2-10) is known to have a zero input solution plus a zero state solution. That is,

$$y(t) = \text{zero input solution} + \int_{t_0}^t h(t-\tau)x(\tau)d\tau \quad (2-11)$$

and, second, the steady-state solution

$$y(t) = \int_{-\infty}^t h(t-\tau)x(\tau)d\tau \quad (2-12)$$

which arises as a limiting case of eqn. (2-11) when all transients have decayed. The forcing function  $x(t)$  is, in this case, assumed to act at all times  $t$ ,  $-\infty < t < \infty$ .

We now derive the Volterra series for eqn. (2-7). We assume that for  $t < 0$ ,  $x(t) = y(t) = 0$ . Equation (2-7) can then be converted into a nonlinear integral equation by applying the inverse operator to  $L_1$ , which, under the assumed initial conditions, is just the following:

$$[L_1(p)]^{-1}x(t) = \int_0^{\infty} h(t-\tau)x(\tau)d\tau \quad (2-13)$$

where  $p = \frac{d}{dt}$ , and  $h(\tau) = 0$  for  $\tau < 0$ . The resulting nonlinear integral



equation is:

$$y(t) + \epsilon \int_0^\infty h(t-\tau) L_2 \left[ \sum_{n=2}^N a_n y^n(\tau) \right] d\tau = \int_0^\infty h(t-\tau) x(\tau) d\tau \quad (2-14)$$

We now assume that  $y(t)$  is given by:

$$y(t) = y_1(t) + y_2(t) + \dots + y_k(t) + \dots \quad (2-15)$$

where  $y_i(t)$  denotes an  $i$ -th order output. Recalling that the order of an output is defined as the number of times the values of the input is multiplied by itself in the convolution integral, we immediately get:

$$y_1(t) = \int_0^\infty h(t-\tau) x(\tau) d\tau \quad (2-16)$$

which is the linear system response. The second order output will be due to the squaring of the first order output. After substituting eqn. (2-15) in eqn. (2-14), we get the following integral equation involving the second-order output:

$$y_2(t) + \epsilon \int_0^\infty h(t-\tau) L_2 [a_2 y_1^2(\tau)] d\tau = 0 \quad (2-17)$$

Substituting for  $y_1(t)$  in eqn. (2-17) from eqn. (2-16), we get

$$y_2(t) = - \epsilon \int_0^\infty h(t-\tau) L_2 [a_2 \int_0^\infty h(\tau-\tau_1) h(\tau-\tau_2) \cdot$$

$$x(\tau_1)x(\tau_2)d\tau_1d\tau_2]d\tau \quad (2-18)$$

$$= - \epsilon a_2 \int_0^\infty \int_0^\infty \int_0^\infty h(t-\tau) \hat{h}(\tau-\tau_1) \hat{h}(\tau-\tau_2) \cdot$$

$$x(\tau_1)x(\tau_2)d\tau_1d\tau_2d\tau \quad (2-19)$$

where  $\hat{h}$  is obtained after the linear operator  $L_2$  has been applied to the quantities within the parenthesis in eqn. (2-18). When  $\epsilon_1 = \epsilon_2 = 0$  in eqn. (2-9), i.e., for the memoryless nonlinear case,  $\hat{h} \equiv \epsilon_3 h$ .

Similarly, when we consider the third-order terms in eqn. (2-14) (which will be due to the second order and first order outputs), we get the following integral:

$$y_3(t) + \epsilon \int_{-\infty}^{\infty} h(t-\tau) L_2 [a_3 y_1^3(\tau) + 2 a_2 y_1(\tau) y_2(\tau)] d\tau = 0 \quad (2-20)$$

Substituting for  $y_1(t)$  and  $y_2(t)$  in eqn. (2-20) and using eqns. (2-16) and (2-19), we get

$$y_3(t) = - \epsilon \int_0^\infty h(t-\tau) [a_3 \int_0^\infty \int_0^\infty \hat{h}(\tau-\tau_1) \hat{h}(\tau-\tau_2) \hat{h}(\tau-\tau_3) \\ + 2 a_2^2 \epsilon \int_0^\infty \int_0^\infty \int_0^\infty h(\tau-\tau_1) h(\tau-\sigma) \hat{h}(\sigma-\tau_2) \cdot$$

$$\hat{h}(\sigma-\tau_3) d\sigma] x(\tau_1) x(\tau_2) x(\tau_3) d\tau_1 d\tau_2 d\tau_3 d\tau \quad (2-21)$$

Higher order outputs can be derived in this bootstrapping manner. We

now give the form of the kernel functions for the nonlinear system described by eqn. (2-7).

When  $n=1$  in eqn. (2-4), we obtain  $h_1(\tau_1)$  from eqn. (2-16) as:

$$h_1(\tau_1) = h(t) = \mathcal{L}^{-1} \left[ \frac{1}{L_1(s)} \right] \quad (2-22)$$

where  $\mathcal{L}^{-1}$  denotes the inverse Laplace transform. It is noted that  $L_1(s)$  is the characteristic equation of the linear system.

Setting  $n=2$  in eqn. (2-4) and comparing\* it with eqn. (2-19), we get

$$h_2(\tau_1, \tau_2) = -\varepsilon a_2 \int_0^\infty h(\tau) \tilde{h}(\tau_1 - \tau) \tilde{h}(\tau_2 - \tau) d\tau \quad (2-23)$$

Similarly, setting  $n=3$  in eqn. (2-4) and comparing with eqn. (2-21), we get:

$$\begin{aligned} h_3(\tau_1, \tau_2, \tau_3) = & -\varepsilon \int_0^\infty h(\tau) [a_3 \tilde{h}(\tau_1 - \tau) \tilde{h}(\tau_2 - \tau) \tilde{h}(\tau_3 - \tau) \\ & + 2a_2^2 \int_0^\infty h(\tau_1 - \tau) h(\sigma - \tau) \tilde{h}(\tau_2 - \sigma) \tilde{h}(\tau_3 - \sigma) d\sigma] d\tau \end{aligned} \quad (2-24)$$

As mentioned previously, the kernel functions are independent of the input. An important feature of the kernel functions given in eqn. (2-23) and (2-24) is that an  $n$ -th order kernel can be expressed as a

\*In eqn. (2-19), we first let  $u = t - \tau$ ,  $u_1 = t - \tau_1$ ,  $u_2 = t - \tau_2$ ; and, therefore,  $\tau_1 = t - u_1$ ,  $\tau_2 = t - u_2$ ,  $\tau - \tau_1 = (t - u) - (t - u_1) = u_1 - u$  and  $\tau - \tau_2 = u_2 - u$ . Then let  $u_1 = \tau_1$ ,  $u_2 = \tau_2$ ,  $u = \tau$  in eqn. (2-23). These steps are worked out in the Appendix.

product of lower order kernels. This product structure property\* holds for many practical lumped system problems. For example, nonlinear circuits with polynomial type nonlinearities have such kernel functions. We will use this property to investigate the convergence properties of Volterra series for this important class of nonlinear system problems.

#### 2-4. Convergence Properties of Volterra Series

The previous section dealt with the question of finding the kernel functions from a nonlinear differential equation. The Volterra series consists of an infinite sum of terms. Clearly, in a practical application, this series must be truncated at some finite term. A very natural question to ask is: How many terms in the series are required to give a good approximation of the system response? Thus, we must look into the convergence properties of Volterra series.

The question of convergence of Volterra series was recently examined by Gilbert [20], and Lesiah [21]. Other references on this subject include Barrett [6,23], Geyer [34], and Ku and Wolf [22]. In keeping with the application-oriented spirit of this dissertation, we first present the basic convergence theorem from [22] and then derive some new and more meaningful parameters, related to the linearized system, that can be used to obtain the approximate error in the truncated series solution.

---

\*Many authors [8] refer to this as the "separable" property of higher order kernels. We use the term "product structure" instead, so as to avoid confusing with separable kernel in linear system theory, which denotes a different concept.

Theorem 2-1 [22]: Let the  $n$ -th order response of the Volterra series (2-2) be given by

$$y_n(t) = \int_0^\infty \cdots \int_0^\infty h_n(\tau_1, \tau_2, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i \quad (2-25)$$

where  $x(t)$  is the input function. Then for a bounded forcing function  $x(t)$ ; i.e., there exists a constant  $X > 0$  such that

$$|x(t)| \leq X \text{ for all } t \quad (2-26)$$

we have

$$|y(t)| = \left| \sum_{n=1}^{\infty} y_n(t) \right| \leq \sum_{n=1}^{\infty} G_n X^n \quad (2-27)$$

where

$$G_n = \int_0^\infty \cdots \int_0^\infty |h_n(\tau_1, \dots, \tau_n)| d\tau_1 \cdots d\tau_n \quad (2-28)$$

Proof: From the triangle inequality, we have

$$|y(t)| = \left| \sum_{n=1}^{\infty} y_n(t) \right| \leq \sum_{n=1}^{\infty} |y_n(t)| \quad (2-29)$$

But,

$$|y_n(t)| = \left| \int_0^\infty \cdots \int_0^\infty h_n(\tau_1, \dots, \tau_n) \prod_{p=1}^n x(t-\tau_p) d\tau_p \right|$$

$$\leq \int_0^\infty \cdots \int_0^\infty |h_n(\tau_1, \dots, \tau_n)| \prod_{p=1}^n |x(t-\tau_p)| d\tau_p$$

$$\leq x^n \int_0^\infty \cdots \int_0^\infty |h_n(\tau_1, \dots, \tau_n)| d\tau_1 \dots d\tau_n \quad (2-30)$$

Using eqns. (2-28) and (2-30) with (2-29) gives

$$|y(t)| \leq \left| \sum_{n=1}^\infty y_n(t) \right| \leq \sum_{n=1}^\infty G_n x^n \quad (2-31)$$

This completes the proof.

Equation (2-31) clearly emphasizes the role which the input amplitude plays in the determination of the number of output terms that must be retained to achieve a desired accuracy. The importance of the input amplitude in the analysis and synthesis of nonlinear can never be over-emphasized. The existence of multiple solutions in nonlinear systems is not rare. Whether the system will exhibit the desired and stable solution very often depends on the input amplitude. Thus the input amplitude plays an important role in the nonlinear system response. Later in this section we shall briefly discuss the stability and boundedness of solutions via the Volterra series method.

Returning to the topic of convergence, it should be clear that eqn. (2-31) is not of much use in its present form; for it requires the determination of the  $n$ -th order kernel before the bound on the  $n$ -th order response can be found. Even when the kernel is determined, the evaluation of  $G_n$  may not be a simple task. In an engineering problem one prefers to have an a priori estimate of the order of the error that

is incurred as a result of truncating a series at some finite term. In the following paragraphs of this sub-section, we explore the convergence of lumped systems with a single nonlinear element that is characterized by the differential equation (2-7). We then investigate the convergence of lumped systems with multiple nonlinear elements.

#### 2-4.1. Single Nonlinear Element Case:

In this section we study a nonlinear differential equation with memoryless nonlinearity and determine the bounds on the solution in terms of the  $L_1$  norm of the linear kernel function, and concomitantly in terms of the poles and residues of the linearized system. The ability to examine the convergence of the solution series in terms of the pole-residue values is of great help to the system analyst, because such information for the linearized system can be easily obtained from many computer-aided analysis packages. Furthermore, the results derived in this section give a means of estimating the bound on the next higher term of the truncated series - something similar to the case of numerical integration formulae, which give the order of the truncation error in terms of the step size.

The equation to be examined is for a nonlinear system with memoryless nonlinearity with quadratic and cubic terms. We examine the following differential equation:

$$L[y(t)] + a_2 y^2(t) + a_3 y^3(t) = x(t) \quad (2-32)$$

where  $L$  is a linear operator, similar to  $L_1$  in eqn. (2-8). Then using eqns. (2-16), (2-19), and (2-21), we can immediately write the approximate solution for  $y(t)$  as:

$$y(t) = \int_0^{\infty} h(t-\tau)x(\tau)d\tau + a_2 \int_0^{\infty} \int_0^{\infty} h(\tau)h(\tau_1-\tau)h(\tau_2-\tau) \cdot$$

$$x(\tau_1)x(\tau_2)d\tau_1d\tau_2d\tau + 2a_2^2 \int_0^{\infty} \int_0^{\infty} \int_0^{\infty} h(\tau)h(\tau_1-\tau)h(\sigma-\tau)$$

$$h(\tau_2-\sigma)h(\tau_3-\sigma)x(\tau_1)x(\tau_2)x(\tau_3)d\tau_1d\tau_2d\tau_3d\sigma d\tau +$$

$$a_3 \int_0^{\infty} \int_0^{\infty} \int_0^{\infty} h(\tau)h(\tau_1-\tau)h(\tau_2-\tau)h(\tau_3-\tau)x(\tau_1)x(\tau_2)x(\tau_3) \cdot$$

$$d\tau_1d\tau_2d\tau_3d\tau + \dots \quad (2-33)$$

Now, if the following conditions hold:

$$|x(t)| \leq X \text{ for all } t \quad (2-34)$$

$$\text{and } \int_0^{\infty} |h(\tau)|d\tau = G; h(\tau) = 0, \tau < 0 \quad (2-35)$$

then the bound on eqn. (2-33) is given by:

$$|y(t)| = Y \leq GX + |a_2|G^3X^2 + 2|a_2|^2G^5X^3 \\ + |a_3|G^4X^3 + \dots \quad (2-36)$$

In fact, if higher order terms were considered, the bound on Y will be given by:



$$Y \leq Y_1 + Y_2 + Y_3 + Y_4 + \dots \quad (2-37)$$

$$Y_1 \leq GX$$

$$Y_2 \leq |a_2|G^3X^2$$

$$Y_3 \leq (2|a_2|^2G^5 + |a_3|G^4)X^3$$

$$Y_4 \leq [(4|a_2|^3 + |a_2|^2)G^7 + (2|a_2||a_3| + 3|a_3|)G^6 + |a_4|G^5]X^4$$

.

.

.

$$Y_i \leq G \left[ \sum_{k=2}^i |a_k| Y_{i,k} \right], \quad i \geq 2 \quad (2-38)$$

where

$$Y_{l,m} = \sum_{j=1}^{l-m+1} Y_j Y_{l-j,m-1}; \quad Y_{l,1} = Y_l \quad (2-39)$$

Equations (2-38) and (2-39) give a recursive relationship for determining the bound on the output terms. As shown by the first few terms, these bounds ultimately depend on the boundedness of the  $l_1$  norm of the kernel function,  $h_1(\tau)$ , of the linearized system and the input,  $x(t)$ .

We now look at an approach to estimate  $G$ . For eqn. (2-32), the transform domain description of an asymptotically stable first order kernel can be written as\*:

$$H(s) = \sum_i \frac{r_i}{s+p_i} + \sum_j \frac{R_j}{[(s+\alpha_j)^2 + \omega_j^2]} \quad (2-40)$$

The impulse response is then given by:

$$h(t) = \sum_i r_i e^{-p_i t} + \sum_j R_j e^{-\alpha_j t} \sin \omega_j t \quad (2-41)$$

From eqn. (2-41), we immediately get

$$|h(t)| \leq \sum_i |r_i| e^{-p_i t} + \sum_j |R_j| e^{-\alpha_j t} |\sin \omega_j t| \quad (2-42)$$

and, therefore,

$$\begin{aligned} G &= \int_0^{\infty} |h(\tau)| d\tau \\ &\leq \sum_i \int_0^{\infty} |r_i| e^{-p_i \tau} d\tau + \sum_j \int_0^{\infty} |R_j| e^{-\alpha_j \tau} |\sin \omega_j \tau| d\tau \end{aligned} \quad (2-43)$$

The first integral in eqn. (2-43) is easily evaluated as follows:

$$G_1 = \sum_i \int_0^{\infty} |r_i| e^{-p_i t} dt$$

---

\*Here we consider the case of linear system with simple poles. The linear system, which is asymptotically stable, but has multiple order poles can be handled similarly.

$$= \sum_i \frac{|r_i|}{|p_i|} \quad (2-44)$$

We now seek a method of estimating the second integral in eqn. (2-43). For this we examine a single complex conjugate pole pair and evaluate the  $j$ -th term in the series [30]. Let

$$g_j(\tau) = R_j e^{-\alpha_j \tau} \sin \omega_j \tau \quad (2-45)$$

Then,

$$\begin{aligned} \int_0^{\infty} |g_j(\tau)| d\tau &= \int_0^{\pi/\omega_j} |g_j(\tau)| d\tau \left[ \sum_{k=0}^{\infty} (e^{-\alpha_j \pi/\omega_j})^k \right] \\ &= \frac{\int_0^{\pi/\omega_j} |g_j(\tau)| d\tau}{1 - e^{-\pi \alpha_j / \omega_j}} \end{aligned} \quad (2-46)$$

Eqn. (2-46) gives a method of evaluating the terms in eqn. (2-43) that involve complex poles. Thus, eqn. (2-43) can be rewritten as:

$$G \leq G_1 + G_2 \quad (2-47)$$

$$\text{where } G_1 = \sum_i \frac{|r_i|}{|p_i|}$$

$$\text{and } G_2 = \sum_j \frac{\int_0^{\pi/\omega_j} |g_j(\tau)| d\tau}{1 - e^{-\pi\alpha_j/\omega_j}} ; g_j(\tau) = R_j e^{-\alpha_j \tau} \sin_j \tau \quad (2-48)$$

Clearly,  $G_1$  is the contribution to  $G$  due to the real poles, and  $G_2$  is the contribution due to the complex poles in the linearized system.

Our discussion in this sub-section has dealt with the convergence of the Volterra series for nonlinear lumped systems with a single memoryless nonlinearity. The results will be summarized in the following theorem, whose proof is included in the appendix.

Theorem 2-2: Consider the equation

$$L\left(\frac{d}{dt}\right) \cdot y(t) + \sum_{i=2}^{\infty} a_i y^i(t) = L_1\left(\frac{d}{dt}\right) \cdot x(t)$$

where  $L(p) = p^m + b_{m-1}p^{m-1} + \dots + b_0$

$$L_1(p) = p^l + c_{l-1}p^{l-1} + \dots + c_0, \quad l < m$$

and  $L(p)$  is assumed to have roots with negative real part. If, for  $t < t_0$ ,  $x(t) = y(t) = 0$ , and for  $t \geq t_0$ ,  $|x(t)| \leq X$ , then  $y(t)$ , for  $t \geq t_0$ , remains bounded by the inequality (2-37).

It should be noted that Theorem 2-2 shows how the series converges; it does not show if the series converges.

The bound on the output, according to eqn. (2-37), ultimately depends on the bound on the  $L_1$  norm of the linear kernel function. This can be easily obtained by the method discussed above for finding  $G$ . If

the series converges, these results can be used to determine the number of terms in the Volterra series that must be retained to get a solution within a prescribed accuracy; one does not have to first determine the kernel functions (except the linear kernel), as per Theorem 2-1, to estimate the number of terms that must be kept. Knowing  $G$  and  $X$  and the coefficients  $a_i$ , the bound on each order output can be determined, which tacitly provides us the information about how many kernels need be determined to meet a prescribed error criterion.

Our foregoing discussion has dealt with nonlinear systems with memoryless nonlinearity. The case of memory type nonlinearity can be treated in a parallel manner, and an inequality similar to eqn. (2-37) derived. The only difference in the new inequality will be that the  $L_1$  norm of the differentiated or integrated first-order kernel function will be involved. Thus, for example, terms of form  $\int_0^\infty \left| \frac{d}{d\tau} h(\tau) \right| d\tau$  or  $\int_0^\infty \left[ \int_0^\tau |h(\lambda)| d\lambda \right] d\tau$  will now be involved in determining the bounds on the various order responses. These terms can again be evaluated easily. For example, consider the case of a system with simple, real poles. Then,

$$h(t) = \sum_i r_i e^{-p_i t}, \quad p_i > 0 \quad (2-49)$$

From (2-49), we get

$$\begin{aligned} \hat{G} &= \int_0^\infty \left| \frac{d}{dt} h(\tau) \right| d\tau = \left| h(\tau) \right|_0^\infty \\ &\leq \sum_i |r_i| \end{aligned} \quad (2-50)$$

We now consider the case of lumped systems with multiple nonlinearities.

#### 2-4.2 Multiple Nonlinear Element Case

The case of lumped systems with multiple nonlinear elements is similar to the case of one nonlinear element. Instead of working with a single differential equation, we work with a system of differential equations. In studying the convergence of the Volterra series we again use the "product structure" property (PS) of the higher order kernel functions to derive the bounds on the various order outputs in terms of the  $l_1$  norm of the linear kernels. Before proceeding with our discussion of the main topic of this sub-section we point out that most higher order kernels encountered in nonlinear systems with polynomial nonlinearities have the PS property\*. In some cases, a "special" viewpoint may be required to see this property.

Consider, for example, the system of Fig. 2-3. Van Trees [8] uses this as an example of a system with non-PS property kernels. His claim follows from looking at the second order input-output relationship, which is:

$$z(t) = \int_0^{\infty} \int_0^{\infty} [h_{1a}(\tau_1)h_{1a}(\tau_2) + h_{1b}(\tau_1)h_{1b}(\tau_2)] \cdot x(t-\tau_1)x(t-\tau_2)d\tau_1d\tau_2 \quad (2-51)$$

---

\*The PS property for cascaded systems is quite apparent from the general expressions for their kernel function. See, for example, [5] to verify this.

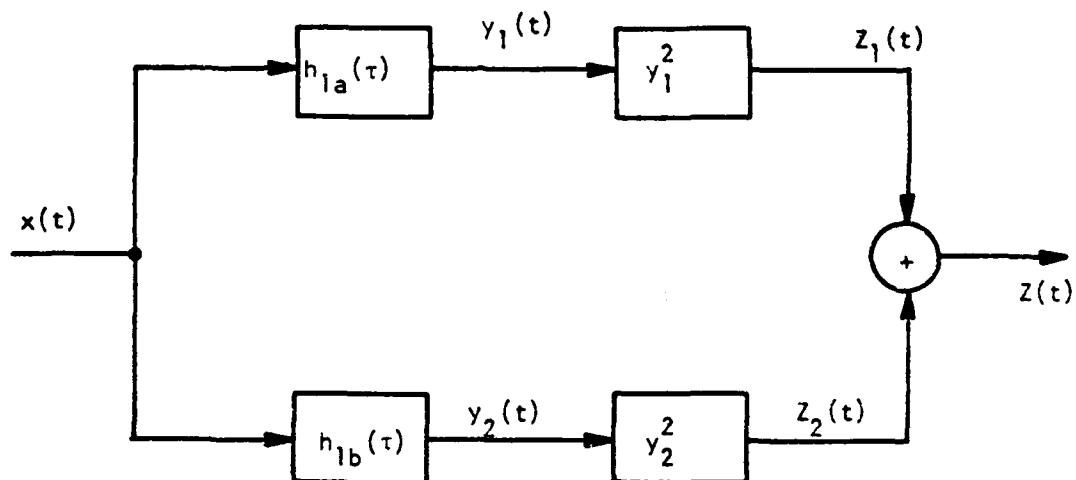


Fig. 2-3. A System with "non-ps" kernels.

However, if we view the output of each nonlinear element separately, and write the system output in terms of these component outputs, then we find that the kernels of the system of Fig. 2-3 indeed have the PS property. This is evident from eqn. (2-51) when we re-write it as:

$$\begin{aligned}
 z(t) &= \int_0^\infty \int_0^\infty h_{1a}(\tau_1) h_{1a}(\tau_2) x(t-\tau_1) x(t-\tau_2) d\tau_1 d\tau_2 \\
 &\quad + \int_0^\infty \int_0^\infty h_{1b}(\tau_1) h_{1b}(\tau_2) x(t-\tau_1) x(t-\tau_2) d\tau_1 d\tau_2 \\
 &= z_1(t) + z_2(t)
 \end{aligned} \tag{2-52}$$

Then, the systems relating  $x(t)$  to  $z_1(t)$  and  $x(t)$  to  $z_2(t)$  have the PS property. Furthermore, the bound,  $Z$ , on  $z(t)$  is given by:

$$Z = |z(t)| \leq |Z_1| + |Z_2| \quad (2-53)$$

where  $Z_1$  and  $Z_2$  are the bounds on the component outputs,  $z_1(t)$  and  $z_2(t)$ , respectively.

We now study the convergence of lumped nonlinear system by working with a specific example. Consider the system described by the following set of equations:

$$\begin{aligned} \ddot{y}_1 + \dot{y}_1 + y_1 + \dot{y}_2 + y_2 + a_2 y_1^2 + a_3 y_1^3 &= x(t) \\ \ddot{y}_2 + \dot{y}_2 + y_2 + y_1 + a_4 y_2^2 + a_5 y_2^3 &= 0 \end{aligned} \quad (2-54)$$

Just as for the case of single equation, we apply the inverse operator to eqns. (2-54) to obtain the integral form of the vector equations:

$$\begin{aligned} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \int_0^t \begin{bmatrix} h_{11}(t-\tau) & h_{12}(t-\tau) \\ h_{22}(t-\tau) & h_{21}(t-\tau) \end{bmatrix} \begin{bmatrix} a_2 y_1^2(\tau) + a_3 y_1^3(\tau) \\ a_4 y_2^2(\tau) + a_5 y_2^3(\tau) \end{bmatrix} d\tau = \\ \int_0^t \begin{bmatrix} h_{11}(t-\tau)x(\tau) \\ h_{21}(t-\tau)x(\tau) \end{bmatrix} d\tau \end{aligned} \quad (2-55)$$

Now we assume that  $y_1(t)$  and  $y_2(t)$  are made up of various order outputs; that is,

$$\begin{aligned} y_1(t) &= y_1^{(1)}(t) + y_1^{(2)}(t) + y_1^{(3)}(t) + \dots \\ y_2(t) &= y_2^{(1)}(t) + y_2^{(2)}(t) + y_2^{(3)}(t) + \dots \end{aligned} \quad (2-56)$$

where  $y_i^{(j)}$  denotes the  $j$ -th order output  $y_i(t)$ . Substituting eqn.



(2-56) in eqn. (2-55), and collecting terms of like order on both sides, the first two terms are found to be:

$$\begin{bmatrix} y_1^{(1)}(t) \\ y_2^{(1)}(t) \end{bmatrix} = \begin{bmatrix} h_{11}(t-\tau)x(\tau)d\tau \\ h_{21}(t-\tau)x(\tau)d\tau \end{bmatrix}$$

$$\begin{bmatrix} y_1^{(2)}(t) \\ y_2^{(2)}(t) \end{bmatrix} = \int_0^{\infty} \tilde{H}(t-\tau) \begin{bmatrix} a_2 \int_0^{\infty} \int_0^{\infty} h_{11}(\tau-\lambda_1)h_{11}(\tau-\lambda_2)x(\lambda_1)x(\lambda_2)d\lambda_1d\lambda_2 \\ a_4 \int_0^{\infty} \int_0^{\infty} h_{21}(\tau-\lambda_1)h_{21}(\tau-\lambda_2)x(\lambda_1)x(\lambda_2)d\lambda_1d\lambda_2 \end{bmatrix} d\tau \quad (2-57)$$

where

$$\tilde{H}(t-\tau) = \begin{bmatrix} h_{11}(t-\tau) & h_{12}(t-\tau) \\ h_{21}(t-\tau) & h_{22}(t-\tau) \end{bmatrix} \quad (2-58)$$

By defining a matrix  $\tilde{G}$  as:

$$\tilde{G} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \triangleq \begin{bmatrix} \int_0^{\infty} |h_{11}(\tau)|d\tau & \int_0^{\infty} |h_{12}(\tau)|d\tau \\ \int_0^{\infty} |h_{21}(\tau)|d\tau & \int_0^{\infty} |h_{22}(\tau)|d\tau \end{bmatrix} \quad (2-59)$$

we can readily determine the bounds on the first and second order outputs, provided the input  $x(t)$  is bounded, as follows:

$$|y_1^{(1)}(t)| = y_1^{(1)} \leq g_{11}x$$

$$|y_2^{(1)}(t)| = y_2^{(1)} \leq g_{21}x \quad (2-60)$$

$$\begin{bmatrix} |y_1^{(2)}(t)| \\ |y_2^{(2)}(t)| \end{bmatrix} = \begin{bmatrix} y_1^{(2)} \\ y_1^{(2)} \end{bmatrix} \leq \tilde{G} \begin{bmatrix} g_{11}^2 x^2 \\ g_{22}^2 x^2 \end{bmatrix} \quad (2-61)$$

The above example of a system with multiple nonlinear elements was deliberately chosen to be simple, and yet we find that the determination of the bounds on the output is quite involved. This example nevertheless illustrates how the bounds on the various order outputs are related to the matrix involving the  $l_1$  norm of the linear kernel functions. We summarize the results by the following theorem.

Theorem 2-3: Consider the system:

$$\tilde{L}\left(\frac{d}{dt}\right) \underline{y}(t) + F[\underline{y}(t)] = \underline{x}(t)$$

where

$$\tilde{L}(p) = \begin{bmatrix} l_{11}(p) & l_{12}(p) & \dots & l_{1n}(p) \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ l_{n1}(p) & l_{n2}(p) & \dots & l_{nn}(p) \end{bmatrix}$$

$$\underline{y}(t) = [y_1(t) \quad y_2(t) \quad \dots \quad y_n(t)]^T$$

$$\underline{x}(t) = [x_1(t) \quad x_2(t) \quad \dots \quad x_n(t)]^T$$

$$F[\underline{y}(t)] = \begin{bmatrix} f_1(y_1) \\ f_2(y_2) \\ \vdots \\ f_n(y_n) \end{bmatrix} = \begin{bmatrix} a_2^{(1)} y_1^2 + a_3^{(1)} y_1^3 + \dots \\ a_2^{(2)} y_2^2 + a_3^{(2)} y_2^3 + \dots \\ \vdots \\ a_2^{(n)} y_n^2 + a_3^{(n)} y_n^3 + \dots \end{bmatrix}$$

with each  $f_{ij}(p)$  has roots with negative real parts. If, for  $t < t_0$ ,  $\underline{x}(t) = \underline{y}(t) = 0$ , and for  $t \geq t_0$ ,  $\max_i |x_i(t)| \leq X$ , then  $\underline{y} \triangleq [|y_1(t)| \quad |y_2(t)| \quad \dots \quad |y_n(t)|]^T$ , for  $t \geq t_0$ , remains bounded by the following inequality:

$$\underline{y} \leq \underline{y}_1 + \underline{y}_2 + \dots$$

where  $\underline{y}_i \triangleq [|y_1^{(i)}(t)| \quad |y_2^{(i)}(t)| \quad \dots \quad |y_n^{(i)}(t)|]^T$   
 = bound on the  $i$ -th order output vector

$$\underline{y}_1 \leq X G \underline{v}$$

$$\underline{y}_i \leq G \underline{y}_{i-1}$$

where

$$\tilde{G} = \begin{bmatrix} \int_0^{\infty} |h_{11}(\tau)| d\tau & \int_0^{\infty} |h_{12}(\tau)| d\tau & \dots & \int_0^{\infty} |h_{1n}(\tau)| d\tau \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ \int_0^{\infty} |h_{n1}(\tau)| d\tau & \int_0^{\infty} |h_{n2}(\tau)| d\tau & \dots & \int_0^{\infty} |h_{nn}(\tau)| d\tau \end{bmatrix}$$

$$\underline{\tilde{Y}}_i = \begin{bmatrix} \sum_{k=2}^i |a_k^{(1)}| \gamma_1^{i,k} \\ \sum_{k=2}^i |a_k^{(2)}| \gamma_2^{i,k} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \sum_{k=2}^i |a_k^{(n)}| \gamma_n^{i,k} \end{bmatrix}$$

$$\gamma_n^{l,m} = \sum_{j=1}^{l-m+1} \gamma_n^{(j)} \gamma_n^{l-j,m-1}; \quad \gamma_n^{l,1} = \gamma_n^{(l)}$$

$$\gamma_n^{(l)} = |y_n^{(l)}(t)| = \text{bound on the } l\text{-th order output, } y_n(t).$$

i-th element of  $\underline{v}$  is 1, if  
 $\underline{v} = [1 \ 1 \ \dots \ 0 \ \dots \ 1]$ ;  $x_i(t)$  is non-zero; otherwise,  
it is zero.

Proof: The proof for the above theorem is similar to that for Theorem 2-2, which is included in the appendix.

Theorem 2-3 gives recursive relationship for determining the bounds on the output series of a class of lumped nonlinear systems. As shown by the example, prior to the theorem, these bounds ultimately depend on the integrable property of the linear kernel functions,  $h_{ij}(\tau)$ . An estimate of the bound of these kernel functions can be obtained by the method outlined in section 2-4.1. The theorem is useful in determining the highest order of kernel functions that are needed to obtain the solution within a prescribed accuracy. Only a knowledge of the G matrix and the bound on the input,  $X$ , are required for this task.

It should be pointed out that the basic idea in the above theorem can be extended to nonlinear systems with multiple independent and dependent (coupled), memoryless or memory-type, nonlinear elements. However, the notation and algebra involved becomes so complicated that it is difficult to put it as a general theorem. One can nevertheless apply the idea of estimating the higher order responses in terms of the bounds on the  $L_1$  norm of the linear kernel functions and the inputs for specific system problems at hand.

### 2-4.3. Stability and Boundedness of Solutions.

Much of the foregoing discussion in this section has dealt with the question of how the Volterra series solution converges, but has ignored the question of whether, if at all, the solution for a system converges. The question of the uniqueness of the solution has also not been addressed thus far. The answers to these later questions are directly related to the stability and boundedness of the solution. By determining the condition on  $X$ , in terms of the  $L_1$  norm of the linear kernel function, for which the Volterra series converges, we can get much insight into the stability and boundedness of a solution. Our discussion in this sub-section will look into the determination of the constraint on  $X$  to get a converging series. Theorem 2-2 will be most useful for such a task.

Much of the theory on the uniqueness and boundedness of solutions of ordinary nonlinear differential equations has been covered in [15,21,35]. In [15] the nonlinear differential equation is converted to the Volterra integral equation (which is an equivalent representation in integral form of eqn. (2-7)), and using the Picard iteration approach, conditions on the nonlinearity, the linear kernel function, and the boundedness of the input, are derived to prove convergence and uniqueness. It is shown that if the forcing function is bounded, the linear kernel is square integrable, and the nonlinearity satisfies the Lipschitz condition, then starting from any set of initial conditions at  $t_0$ , (2-7) has a unique solution for any finite range of time  $t$ . The proof fails if either  $t_0 \rightarrow -\infty$  or  $t \rightarrow \infty$ . Since it has been shown [19] that the sequence of iterates, obtained from the iteration process,

correspond to the partial sums of the Volterra series, it follows that the above criterion for uniqueness and boundedness can also be applied when the Volterra series method is used. However, if one is using the Volterra series method, it is only natural to have some criterion which establishes a direct relationship between the convergence of the series and the stability and boundedness of the solutions. Most authors [20,21] have established conditions, similar to those stated above, to prove convergence and uniqueness of the Volterra series in a finite time interval  $[0,T]$ ; the case of  $[0,\infty]$ , or  $[-\infty,\infty]$  has not been addressed. The latter problem has not been adequately dealt with in the literature and is in fact a difficult problem. The only known work in this area was done by Barrett [23]. We use results from [23] to a specific equation to show the possibility of determining the stability and boundedness via the Volterra series method.

Consider the following second-order nonlinear differential equation with a cubic nonlinearity:

$$\ddot{y} + a_1 \dot{y} + a_0 y + \epsilon y^3 = x(t) \quad (2-61)$$

Then using theorem 2-2, we can immediately write the bound on the solution  $y(t)$  as:

$$|y(t)| = Y \leq GX + |\epsilon| G^4 X^3 + \dots \quad (2-62)$$

where

$$G \triangleq \int_0^{\infty} |h(\tau)| d\tau ; h(t) = \mathcal{L}^{-1} \left[ \frac{1}{s^2 + a_1 s + a_0} \right] \quad (2-63)$$

$$X \triangleq \max_{0 \leq t < \infty} |x(t)|$$

Re-writing (2-62) as:

$$Y = f(X) = GX + |\epsilon| G^4 X^3 + \dots \quad (2-64)$$

It should be evident that if the series for  $f(X)$  in eqn. (2-64) converges for some  $X \leq X_c$ , then the Volterra series solution for eqn. (2-61) also converges for  $\max_{0 \leq t < \infty} |x(t)| < X_c$ , since  $f(X)$  dominates the Volterra series term by term. We now seek a nonlinear algebraic equation for which the series  $f(X)$  is a solution. The roots of this algebraic equation along with the constraints on  $X$  for which the series is convergent will give more insight into the solution of eqn. (2-61). All this will become evident as we proceed along.

It can be shown that the nonlinear algebraic equation\* for which  $f(X)$  is a solution is:

$$Y - |\epsilon| G Y^3 = GX \quad (2-65)$$

Being a cubic, eqn. (2-65) will have three solutions for  $Y$  in terms of  $X$ . This shown in Fig. 2-4, where

---

\*After the inverse operator is applied to eqn. (2-61), there results a striking resemblance with eqn. (2-65).



$$X_c = \frac{2}{\sqrt{|\epsilon|} (3G)^{3/2}} \quad (2-66)$$

$$Y_c = \frac{1}{\sqrt{3}|\epsilon|G} \quad (2-67)$$

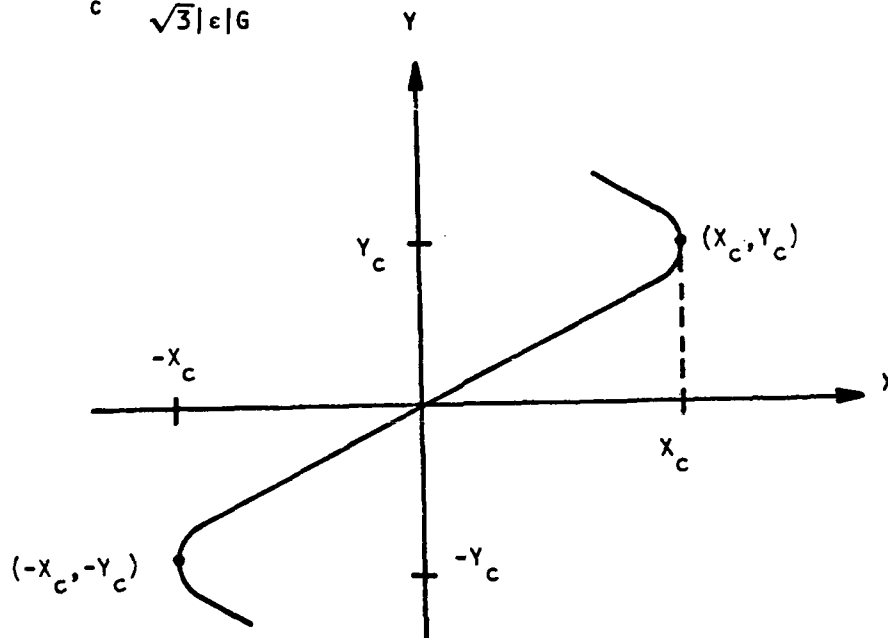


Fig. 2-4. Roots of the Nonlinear Algebraic Equation

Figure 2-4 provides some interesting insight into the behavior of the solutions of eqn. (2-65), and concomitantly into the solutions of eqn. (2-61). First, when  $X \rightarrow 0$ , then  $Y = GX$  - which implies a "linear system" operation. When  $X < X_c$ , all three solutions of eqn. (2-65) are given by\*:

$$Y = 3X_c G \sin \left[ \frac{1}{3} \sin^{-1} \frac{X}{X_c} \right] \quad (2-68)$$

The range of convergence for the solution  $Y$  in eqn. (2-68) is that

\*If  $\sin^{-1}$  is given its principal value, the relation  $Y = f(X)$  results.

$|x| < X_c$ , where  $X_c$  is defined in terms of  $G$  and  $\epsilon$ . Since the algebraic equation (2-64) dominates the Volterra series solution to eqn. (2-61) term by term, it follows that the series for eqn. (2-61) will also converge provided  $|x(t)| < X_c$ , for all  $t$ , and also remain bounded by the inequality (2-62).

The above example illustrates an interesting method for arriving at the constraint on the input function in order to insure that the Volterra series solution converges. Instead of working with the functional series, we work with a nonlinear algebraic equation. The behavior of the roots of the algebraic equation in a given region then provides insight into the convergence of Volterra series.

The above approach will also show the cases in which the Volterra series method will not yield meaningful results. Although not proven yet, it appears that if the linearized system is not asymptotically stable, then the Volterra series solution may not converge. This conjecture can be seen when we consider the effect of a simple pole which is very close to the  $j\omega$ -axis. This pole will predominantly effect the  $l_1$  norm of the linear kernel function. If  $X_c$  has  $G$  in the denominator, as in the above illustration, it should be clear that as  $G \rightarrow \infty$ ,  $X_c \rightarrow 0$ . Thus, the bound on the input for which the series converges will go to zero as the pole moves very close to the  $j\omega$ -axis.

## CHAPTER 3

### MULTIPLE-NODE, MULTIPLE-NONLINEARITY CIRCUIT ANALYSIS

#### 3-1. Introduction

Nonlinear systems that admit a Volterra series description are completely characterized by their nonlinear impulse response functions or the generalized transfer functions, which are the multi-dimensional transforms of the nonlinear impulse response functions. Thus, any analysis of nonlinear systems via the Volterra series method will entail the determination of either one of these functions.

Previous works [7,10,12,13,14,43] for determining the generalized transfer functions have relied on the "harmonic input" method or the "nonlinear current source" method. The harmonic input method involves the use of exponential functions as inputs, equating terms of like order on both sides of the system equation(s), and then determining the transfer functions. The nonlinear current source method involves the solving of the nonlinear differential equation(s) by repeatedly solving a linear differential equation with nonlinear excitation, again assuming harmonic inputs. Both these methods give a recursive relationship between an  $n$ th order transfer function and up to the  $(n-1)$ -order transfer functions. Although both methods are mathematically correct and yield the correct generalized transfer functions for a system, they both have a common "drawback": there is no direct relationship between the approach used and the definition of the generalized transfer functions.

Since the generalized transfer functions are transforms of the impulse response functions, it only stands to reason that the determination of these transfer functions should somehow entail the use of an impulse function as an input to the system - just as in the case of linear systems. Based on this intuitive approach and multi-dimensional transform theory, we develop a new method for the determination of generalized transfer functions which does not involve the use of exponential functions as inputs.

As one would expect, the results obtained from the approach developed here coincide with those obtained previously. However, it is believed that this new approach has some advantages: first, the approach relies on fundamental definitions of multi-variable transforms, and is therefore more rigorous than the "heuristic" frequency probing methods; secondly, the morass of algebra involved in the harmonic input method is avoided; third, the new approach is more intuitive and provides a direct link to the work done elsewhere [5] on nonlinear systems.

In this section, we develop this new approach gradually. In section 3-2 the multi-dimensional transform theory is introduced, along with the application of the theory to specific examples which will be subsequently used in deriving the generalized transfer functions. In section 3-3 the generalized transfer functions for an  $r$ -th order nonlinear differential equation are obtained. Section 3-4 is devoted to the determination of the nonlinear transfer functions of a general multiple-node, multiple-nonlinearity circuit with a single input. The case of multiple input sources is treated in section 3-5. A specific

nonlinear circuit example is presented in section 3-6.

### 3-2. Multi-dimensional Transforms

The Laplace transform pair of a one-dimensional function,  $f(t)$ , is:

$$F(s) = \int_{-\infty}^{\infty} f(t) e^{-st} dt \quad (3-1)$$

and

$$f(t) = \frac{1}{(2\pi j)} \int_{\sigma-j\infty}^{\sigma+j\infty} F(s) e^{st} ds \quad (3-2)$$

For a multi-variable function,  $f(t_1, t_2, \dots, t_n)$ , the corresponding multi-dimensional transform [24] is:

$$F(s_1, s_2, \dots, s_n) = \int \dots \int_{n\text{-fold}} f(t_1, t_2, \dots, t_n) \exp(-s_1 t_1 - \dots - s_n t_n) dt_1 \dots dt_n \quad (3-3)$$

and

$$f(t_1, \dots, t_n) = \frac{1}{(2\pi j)^n} \int \dots \int_{n\text{-fold}} F(s_1, \dots, s_n) \exp(s_1 t_1 + \dots + s_n t_n) ds_1 \dots ds_n \quad (3-4)$$

$$f(t_1, \dots, t_n) \leftrightarrow F(s_1, \dots, s_n) \quad (3-5)$$

Before proceeding further, we make the following notational definitions:

$$F(s_1, s_2, \dots, s_n) = \mathcal{L}[f(t_1, t_2, \dots, t_n)] \quad (3-6)$$

and

$$f(t_1, t_2, \dots, t_n) = \mathcal{L}^{-1}[F(s_1, s_2, \dots, s_n)] \quad (3-7)$$

Whether we use Fourier transform or Laplace transform in eqns. (3-3) and (3-4) depends on the contours of integration and values of  $s_1, s_2, \dots, s_n$ . The importance of the region of convergence when dealing with unstable and non-causal linear systems is well known. Here we assume that the systems under consideration are causal; that is, the Volterra kernels  $h_n(t_1, t_2, \dots, t_n) = 0$ , for  $t_1, t_2, \dots, t_n \leq 0$ . Also, in general, we are concerned with functions (or generalized functions) whose region of convergence includes the imaginary axis in each variable, so that the Fourier transform is included in our definitions.

It should also be noted that most of the properties of the one-dimensional transform (linear case) carry over to the multi-dimensional case. The validity of this statement can be checked elsewhere [5].

It is often desirable to express the multi-variable function,  $f(t_1, t_2, \dots, t_n)$ , as a simple function of time,  $f(t)$ , and vice versa. If all  $t_i$ 's are restricted to be identical so that  $t = t_1 = t_2 = \dots = t_n$ , then  $f(t_1, t_2, \dots, t_n)$  becomes  $f(t)$ . Thus, in the two variable case,  $f(t)$  can be obtained from  $f(t_1, t_2)$  by evaluating  $f(t_1, t_2)$  along the 45° line  $t_1 = t_2$ . Similarly, if we plot  $f(t_1, t_2, t_3)$  in a three-dimensional space, then, to obtain  $f(t)$ , we are only interested in  $f(t_1, t_2, t_3)$  along the line  $t_1 = t_2 = t_3$ . The idea of converting a nonlinear function of one variable  $t$  into a product of linear multi-variable functions will be used repeatedly in the sequel. One must, however, bear in mind that the ultimate goal is to obtain the solution of the differential equation as a function of time,  $t$ , and that the introduction of  $t_1, t_2$ , etc. are merely for mathematical manipulations.

We now apply multi-dimensional transforms to some specific cases which will be subsequently used in sections (3-3) and (3-4).

**3-2.1 Volterra Series:** The Volterra series relates the system input  $x(t)$  to the system  $y(t)$  as follows\*:

$$\begin{aligned}
 y(t) &= \sum_{n=1}^{\infty} \int \cdots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i \\
 &= \sum_{n=1}^{\infty} y_n(t)
 \end{aligned} \tag{3-8}$$

where

$$y_n(t) = \int \cdots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i \tag{3-9}$$

Introducing dummy variables  $t_1, t_2, \dots, t_n$  in eqn. (3-9) we can write  $y_n(t)$  as:

$$\begin{aligned}
 y_n(t) &= y_n(t_1, t_2, \dots, t_n) |_{t_1=t_2=\dots=t_n=t} \\
 &= \int \cdots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t_i - \tau_i) d\tau_i
 \end{aligned} \tag{3-10}$$

Taking the  $n$ -dimensional transforms of eqn. (3-10), we get:

---

\*Unless otherwise stated, all limits of integration are between 0 and  $\infty$  in our discussion here.

$$Y_n(s_1, \dots, s_n) = \mathcal{L}[y_n(t_1, \dots, t_n)]$$

$$= \int \dots \int_{2n\text{-fold}} h_n(\tau_1, \tau_2, \dots, \tau_n) \prod_{i=1}^n x(t_i - \tau_i) e^{-s_i t_i} d\tau_i dt_i \quad (3-11)$$

Defining  $t_n - \tau_n = \sigma_n$ ,  $t_{n-1} - \tau_{n-1} = \sigma_{n-1}$ , ...,  $t_1 - \tau_1 = \sigma_1$ , and therefore:  
 $t_n = \sigma_n + \tau_n$ ,  $t_{n-1} = \sigma_{n-1} + \tau_{n-1}$ , ...,  $t_1 = \sigma_1 + \tau_1$ ;  
 $d\sigma_n = dt_n$ ,  $d\sigma_{n-1} = dt_{n-1}$ , ...,  $d\sigma_1 = dt_1$ . Substituting these quantities  
in eqn. (3-11) and performing the 2n-fold integrations with respect to  
 $d\tau_i$  and  $d\sigma_i$ , we get

$$Y_n(s_1, \dots, s_n) = H_n(s_1, \dots, s_n) \prod_{i=1}^n X(s_i) \quad (3-12)$$

and therefore the transform domain description of eqn. (3-8) becomes:

$$Y(s_1, s_2, \dots, s_n) = \sum_{n=1}^{\infty} H_n(s_1, \dots, s_n) \prod_{i=1}^n X(s_i) \quad (3-13)$$

where  $X(s)$  is the transform of  $x(t)$ . If the input  $x(t)$  is a delta function, then eqns. (3-12) and (3-13) reduce, respectively, to:

$$Y_n(s_1, \dots, s_n) = H_n(s_1, s_2, \dots, s_n) \quad (3-14)$$

and

$$Y(s_1, \dots, s_n) = \sum_{n=1}^{\infty} H_n(s_1, \dots, s_n) \quad (3-15)$$

Equations (3-12) through (3-14) will be used repeatedly in section (3-3).



3-2.2 Nonlinear Terms. The characteristics of nonlinear elements encountered in many nonlinear dynamical systems can be represented over any finite range by a polynomial. This gives rise to nonlinear differential equations with polynomial type nonlinear terms. When such elements are used in a system, the equilibrium equations contain integrals and derivatives of the polynomials. In this section we apply multi-dimensional transforms to these nonlinear terms to obtain general forms of the transform-domain description of these terms. The details of some of these derivations are given in Appendix B.

$y^2(t)$  Term:

$$y^2(t) = y(t_1)y(t_2)|_{t_1=t_2=t} \quad (3-16)$$

$$\begin{aligned} Y(s_1, s_2) &= \mathcal{L}[y^2(t)] = \iint y(t_1)y(t_2)e^{-s_1t_1-s_2t_2}dt_1dt_2 \\ &= Y(s_1)Y(s_2) \end{aligned} \quad (3-17)$$

$y^3(t)$  Term:

$$y^3(t) = y(t_1)y(t_2)y(t_3)|_{t_1=t_2=t_3} \quad (3-18)$$

$$\begin{aligned}
 Y(s_1, s_2, s_3) &= \mathcal{L}[y^3(t)] = \iiint y(t_1)y(t_2)y(t_3)e^{-s_1t_1-s_2t_2-s_3t_3}dt_1dt_2dt_3 \\
 &= Y(s_1)Y(s_2)Y(s_3)
 \end{aligned} \tag{3-19}$$

$y^n(t)$  Term:

$$y^n(t) = \prod_{i=1}^n y(t_i) \quad t_i = t \tag{3-20}$$

and

$$Y(s_1, \dots, s_n) = \mathcal{L}[y^n(t)] = \prod_{i=1}^n Y(s_i) \tag{3-21}$$

$\frac{d}{dt}y^2(t)$  Term:

$$\frac{d}{dt}y^2(t) = \frac{d}{dt}y(t_1, t_2)|_{t_1=t_2=t} ; y(t_1, t_2) = y(t_1)y(t_2)$$

$$Y(s_1, s_2) = \mathcal{L}\left[\frac{d}{dt}y^2(t)\right] = \iint \frac{d}{dt}y(t_1, t_2)e^{-s_1t_1-s_2t_2}dt_1dt_2$$

$$= (s_1 + s_2)Y(s_1, s_2)$$

$$= (s_1 + s_2)Y(s_1)Y(s_2) \tag{3-22}$$

$\frac{d}{dt}y^n(t)$  Term:

$$\frac{d}{dt}y^n(t) = \frac{d}{dt}y(t_1, \dots, t_n) \big|_{t_1=t_2=\dots=t_n} ; y(t_1, \dots, t_n) = \prod_{p=1}^n y(t_p) \quad (3-23)$$

$$\begin{aligned} Y(s_1, s_2, \dots, s_n) &= \mathcal{L} \left[ \sum_{s=1}^n \frac{\partial}{\partial t_s} y(t_1, \dots, t_n) \frac{dt_s}{dt} \right] \\ &= (s_1 + s_2 + \dots + s_n) Y(s_1) \dots Y(s_n) \end{aligned} \quad (3-24)$$

$\int y^n(t) dt$  Term:

$$\int y^n(t) dt = \int y_n(\tau_1 - t, \tau_2 - t, \dots, \tau_n - t) dt \quad (3-25)$$

Letting  $\tau_i - t = t_i$ , and taking the transform of eqn. (3-25), we get

$$Y(s_1, s_2, \dots, s_n) = Y_n(s_1, \dots, s_n) / (s_1 + s_2 + \dots + s_n) \quad (3-26)$$

$$= \left[ \frac{1}{s_1 + s_2 + \dots + s_n} \right] \prod_{i=1}^n Y(s_i) \quad (3-27)$$

The general forms in eqns. (3-21), (3-24) and (3-27) will be used in sections (3-3) and (3-4). The salient feature in each of these equations is how an nth degree polynomial function in the time-domain is represented by the nth-order product of the transform of the function in the transform domain. It is this product structure which, analogous to the case of linear system analysis, makes the analysis of nonlinear sys-

tems easier via the transform-domain approach.

### 3-3. A Nonlinear Differential Equation:

In this section, we present a method, based on applying the multi-dimensional transforms to nonlinear differential equations, to determine the response of a nonlinear system with a functional power series type of nonlinearity. The nonlinear differential equation considered is the following:

$$L_1[y(t)] + L_2\left[\sum_{n=2}^N a_n y^n(t)\right] = x(t) \quad (3-28)$$

where  $x(t)$  and  $y(t)$  are system input and output, respectively,  $L_1$  is a linear differential operator:

$$L_1[\cdot] = \sum_{r=0}^R \frac{d^r}{dt^r} [\cdot] \quad (3-29)$$

and  $L_2$  is  $\frac{d}{dt}$ ,  $\int$ , or a constant, or a sum of these operators. It should be noted that the linear operator,  $L_2$ , operates on a polynomial function of  $y(t)$ .

We now present an approach whereby the nonlinear differential equation (3-28) is solved by a bootstrapping operation by first dissolving it into a set of linear differential equations with nonlinear inputs. Multidimensional transforms are then applied to these new equations to obtain the Volterra series solution.

There are many different methods of rendering a nonlinear differential equation into a set of linear differential equation with nonlinear inputs. We use the approach outlined in [14,17].

Assume that the input in eqn. (3-28) is of the form

$$x(t) = \epsilon v(t) \quad (3-30)$$

The dummy variable  $\epsilon$  helps to keep track of the order of the terms: a term with coefficient  $\epsilon^n$  signifies an  $n$ th order term. This can be seen easily by substituting eqn. (3-30) in eqn. (3-9), which yields:

$$y_n(t) = \epsilon^n \int \dots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n v(t-\tau_i) d\tau_i \quad (3-31)$$

Let us assume that  $r(t)$  is the response to the input  $v(t)$  in eqn (3-28). Then, according to the Volterra series expansion, as per eqn. (3-8) and (3-9), the  $n$ -th order response is:

$$r_n(t) = \int \dots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n v(t-\tau_i) d\tau_i \quad (3-32)$$

Comparing (3-32) and (3-31), we obtain the following relationships:

$$y_n(t) = \epsilon^n r_n(t) \quad (3-33)$$

and therefore, as per eqn. (3-8),

$$y(t) = \sum_{n=1}^{\infty} y_n(t) = \sum_{n=1}^{\infty} \epsilon^n r_n(t) \quad (3-34)$$

We now have two differential equations which relate  $r(t)$  and  $v(t)$ . First, equation (3-28) can be re-written as:

$$L_1[r(t)] + L_2\left[\sum_{n=2}^N a_n r^n(t)\right] = v(t) \quad (3-35)$$

second, after substituting eqn. (3-34) into (3-28), we get:

$$L_1\left[\sum_{n=1}^{\infty} \epsilon^n r_n(t)\right] + L_2\left[\sum_{j=2}^N (a_j \sum_{n=1}^{\infty} \epsilon^n r_n(t))^j\right] = \epsilon v(t) \quad (3-36)$$

Thus in order to solve eqn. (3-28), we can solve eqn. (3-36) for  $r_n(t)$ ,  $n = 1, 2, \dots$  and substitute in eqn. (3-34) to solve for  $y(t)$  after setting  $\epsilon = 1$ . Setting  $\epsilon = 1$  implies that  $x(t) = v(t)$ , and therefore  $y(t) = \dot{r}(t) = \sum_n \dot{r}_n(t)$ . The introduction of  $\epsilon$  is a mathematical artifice which helps to equate coefficients of  $\epsilon^n$  on both sides of eqn. (3-36), thereby yielding linear differential equations (involving successively higher order outputs) with nonlinear inputs. This is similar to the perturbation method, [17,29].

To solve for  $r_1(t)$ , the linear system response, we equate coefficients of  $\epsilon^1$  on both sides of eqn. (3-36), thus yielding the following equation:

$$L_1[r_1(t)] = v(t) \quad (3-37)$$

Similarly we equate coefficients of  $\epsilon^2, \epsilon^3, \epsilon^4, \epsilon^5$ , and so on, on both sides of eqn. (3-36) to obtain the following equations:

$$L_1[r_2(t)] + L_2[a_2 r_1^2(t)] = 0 \quad (3-38)$$

$$L_1[r_3(t)] + L_2[2a_2 r_1(t)r_2(t) + a_3 r_1^3(t)] = 0 \quad (3-39)$$

$$L_1[r_4(t)] + L_2[a_2(2r_1(t)r_3(t) + r_2^2(t)) + 3a_3r_1^2(t)r_2(t) + a_4r_1^4(t)] = 0 \quad (3-40)$$

$$L_1[r_5(t)] + L_2[2a_2r_1(t)r_4(t) + a_3(3r_1^2(t)r_3(t) + 3r_1(t)r_2^2(t)) + 4a_4r_1^3(t)r_2(t) + a_5r_1^5(t)] = 0 \quad (3-41)$$

•  
•  
•

To solve for the generalized transfer functions of eqn. (3-35), we take the 1-dimensional transform of eqn. (3-37) and obtain:

$$L_1(s_1)R_1(s_1) = V(s_1) \quad (3-42)$$

If  $v(t) = \delta(t)$ , then  $V(s_1) = 1$ , and therefore, according to eqn. (3-14), we have

$$R_1(s_1) = H_1(s_1) = \frac{1}{L_1(s_1)} \quad (3-43)$$

To solve for the second-order transfer functions,  $H_2(s_1, s_2)$ , we perform a 2-dimensional transform of eqn. (3-38) to obtain

$$L_1(s_1+s_2)R_2(s_1, s_2) + a_2L_2(s_1+s_2)R_1(s_1)R_1(s_2) = 0 \quad (3-44)$$

Notice that we have used eqn. (3-21) to transform the  $r_1^2(t)$  term. Using (3-14) and (3-43) in eqn. (3-44), we obtain

$$R_2(s_1, s_2) = H_2(s_1, s_2) = - \frac{a_2 L_2(s_1 + s_2) H_1(s_1) H_1(s_2)}{L_1(s_1 + s_2)} \quad (3-45)$$

Similarly, taking a 3-dimensional transform of eqn. (3-39), we obtain

$$\begin{aligned} L_1(s_1 + s_2 + s_3) R_3(s_1, s_2, s_3) + L_2(s_1 + s_2) [2a_2 R_1(s_1) R_2(s_2, s_3) \\ + a_3 R_1(s_1) R_1(s_2) R_1(s_3)] = 0 \end{aligned} \quad (3-46)$$

Again, using eqns. (3-43), (3-45), and (3-14), we get

$$\begin{aligned} R_3(s_1, s_2, s_3) = H_3(s_1, s_2, s_3) = - L_2(s_1 + s_2 + s_3) [2a_2 H_1(s_1) H_2(s_2, s_3) \\ + a_3 H_1(s_1) H_1(s_2) H_1(s_3)] / L_1(s_1 + s_2 + s_3) \end{aligned} \quad (3-47)$$

In a similar manner, we can derive by inspection:

$$\begin{aligned} H_4(s_1, s_2, s_3, s_4) = - L_2 \left( \sum_{i=1}^4 s_i \right) [a_2 (2H_1(s_1) H_3(s_2, s_3, s_4) \\ + H_2(s_1, s_2) H_2(s_3, s_4)) + 3a_3 H_1(s_1) H_1(s_2) H_2(s_3, s_4) \\ + a_4 \prod_{i=1}^4 H_1(s_i)] / L_1 \left( \sum_{i=1}^4 s_i \right) \end{aligned} \quad (3-48)$$

and

$$H_5(s_1, s_2, s_3, s_4, s_5) = - L_2 \left( \sum_{i=1}^5 s_i \right) [2a_2 H_1(s_1) H_4(s_2, s_3, s_4, s_5)$$



$$\begin{aligned}
& + \overline{3a_3(H_1(s_1)H_1(s_2)H_3(s_3,s_4,s_5))} \\
& + \overline{H_1(s_1)H_2(s_2,s_3)H_2(s_4,s_5))} \\
& + \overline{4a_4H_1(s_1)H_1(s_2)H_1(s_3)H_2(s_4,s_5)} + a_5 \prod_{i=1}^5 H_1(s_i) ] / L_1 \left( \sum_{i=1}^5 s_i \right) \quad (3-49)
\end{aligned}$$

The overbars in eqn. (3-47) to (3-49) represents symmetrization operation, which is an averaging operation. In general, the generalized transfer functions are not symmetrical in their arguments; that is,  $H_2(s_1, s_2)$  may not be equal to  $H_2(s_2, s_1)$ . The symmetrization operation on an unsymmetrical system is performed by summing each of the  $n$ th order transfer function over all permutations of its arguments and dividing by the number of components in the sum.

The use of symmetric transfer functions is not merely for notational convenience, but is necessitated by the method we use for introducing the parameters  $t_1, t_2, \dots$ , before taking the transforms. To illustrate this, we note that  $v_1(t)v_2(t)$  can be written as  $v_1(t_1)v_2(t_2, t_3)$ ,  $v_1(t_2)v_2(t_1, t_3)$ , or  $v_1(t_3)v_2(t_1, t_2)$ . The first term has transform:  $V_1(s_1)V_2(s_2, s_3)$ ; the second term has:  $V_1(s_2)V_2(s_1, s_3)$ ; and the third has transform:  $V_1(s_3)V_2(s_1, s_2)$ . When  $V_2(\cdot, \cdot)$  is not symmetrical in its arguments, each transformed quantity above will yield a different value. Thus, it becomes necessary to use symmetric transfer functions when performing numerical computations to obtain the system response. It can be shown that the response is unchanged when symmetrized transfer functions are used. For example, expressing

$$v_1(t)v_2(t) = \frac{1}{3} [v_1(t_1)v_2(t_2, t_3) + v_1(t_2)v_2(t_1, t_3) + v_1(t_3)v_2(t_1, t_2)],$$

and recalling that  $t_1 = t_2 = t_3 = t$ , does not change the contribution due to  $v_1(t)v_2(t)$  in the system response. In the remaining part of this report we will assume the generalized transfer functions to be symmetric in their arguments.

To conclude this sub-section, we summarize the approach for obtaining the generalized transfer functions of a nonlinear system and also comment on the important ramification of the method. By introducing a dummy variable in the nonlinear differential equation characterizing the system, a set of differential equations of the following form was obtained:

$$L[r_n(t)] + f(r_{n-1}(t)) = 0, \quad n = 2, 3, \dots \quad (3-50)$$

where  $L$  is the linear system operator and  $f(\cdot)$  is a nonlinear function of  $r_{n-1}(t), r_{n-2}(t), \dots, r_1(t)$ .  $r_1(t)$  is the first-order response, which is simply the response of the linear system. The relationship in eqn. (3-50) is clearly a recursive one, and can be used to solve for  $r_n(t)$  in terms of  $r_{n-1}(t), r_{n-2}(t)$ , etc. This is done by taking the  $n$ -dimensional transform of eqn. (3-50) to solve for  $R_n(s_1, \dots, s_n)$ , which is identically the  $n$ th-order transfer function when the input  $v(t)$  is an impulse function. The transform of  $f(\cdot)$  is done by inspection with the help of the results of section (3-2). The  $n$ -dimensional transform of  $L[r_n(t)]$  is shown to be  $L(s_1 + s_2 + \dots + s_n)R_n(s_1, s_2, \dots, s_n)$ . With all this information, eqn. (3-50) is easily solved for the generalized transfer functions.

Much of the discussion above has been concerned with the transform-domain description of the nonlinear system. It should be

pointed out that the form of eqns. (3-37) to (3-41) is quite suitable for obtaining the time domain kernels of the system. By defining an operator  $L_1^{-1}$ , we can easily solve for  $h_1(\tau_1)$ . Knowing  $h_1(\tau_1)$ , we can successively solve for  $h_2(\tau_1, \tau_2)$ ,  $h_3(\tau_1, \tau_2, \tau_3)$ , etc., yielding\*:

$$h_1(t_1) = K(t_1) \quad (3-51)$$

$$h_2(\tau; t_1, t_2) = a_2 \int K(\tau) h_1(t_1 - \tau) h_1(t_2 - \tau) d\tau \quad (3-52)$$

$$\begin{aligned} h_3(\tau; t_1, t_2, t_3) = & \int K(\tau) [a_3 h_1(t_1 - \tau) h_1(t_2 - \tau) h_1(t_3 - \tau) \\ & + 2a_2 h_1(t_1 - \tau) h_2(\tau_1 - \tau; t_2, t_3)] d\tau \end{aligned} \quad (3-53)$$

$$\begin{aligned} h_4(\tau; t_1, t_2, t_3, t_4) = & - \int K(\tau) [a_2 (2h_1(t_1 - \tau) h_3(\tau_1 - \tau; t_2, t_3, t_4) \\ & + h_2(\tau_2 - \tau; t_1, t_2) h_2(\tau_3 - \tau; t_3, t_4)) + \\ & 3a_3 h_1(t_1 - \tau) h_1(t_2 - \tau) h_2(\tau_4 - \tau; t_3, t_4) \\ & + a_4 \prod_{i=1}^4 h_1(t_i - \tau)] d\tau \end{aligned} \quad (3-54)$$

with

$$K(t) \triangleq L_1^{-1}[1/L_1(s)] \quad (3-55)$$

The fifth-order kernel can also be written by inspection of eqn. (3-41).

\*Here we have assumed  $L_2$  to be a "constant" multiplication operator, thus giving a polynomial type memoryless nonlinearity.

However, for the sake of space, we do not write it here. We note here that there is a one-to-one correspondence of terms between the time-domain and the transform-domain description of the kernels as presented above. Thus, knowing the transform-domain functions, the time-domain kernels can be obtained directly using the above equations. The procedure must start from the first order kernel and proceed successively upwards to determine the second-order, third-order,...,n-th order kernels.

#### 3-4. Multiple-Node, Multiple-Nonlinearity Circuit Analysis

Many analysis and design problems in circuits and systems involve one or at most a few nonlinear elements in an otherwise linear time-invariant circuit or system. When a single nonlinear element is present, the differential equation (3-28) and the material of section (3-2) will be adequate for analyzing the nonlinear circuit. For, in such a case, the linear circuit can be characterized by a convolution kernel (via the Thevenin or Norton Theorems) to give the overall Volterra integral equation [15], which can also be cast in a differential equation form, similar to eqn. (3-28).

However, when multiple nonlinear elements are imbedded in an otherwise linear time-invariant circuit, the analysis entails the solution of a system of nonlinear differential equations. The approach developed in section (3-2) for the scalar case is still applicable, but must be extended to solve the system of nonlinear differential equations.

The number of equations to be solved depends on the number and the type of nonlinear elements considered. When only independent type nonlinear elements are considered, the number of equations is less than or

equal to the number of nonlinear elements (assuming that the output is across one of the nonlinear elements; otherwise, an extra equation relating the nonlinear element voltages (currents) and the output voltage (current) is needed to solve for the output). The nonlinear differential equations in such a case is again derived by obtaining the Thevenin (Norton) equivalent circuit (for the linear part of the nonlinear circuit) at each of the ports at which the nonlinear elements are present. When dependent type nonlinear elements are also allowed, the analysis becomes more complicated; for, in such a case, the controlling variables, which may be across a linear element, must be solved for and substituted in the differential equation for the nonlinear element.

Previous works [7,10,14] for determining the generalized voltage ratio transfer functions of lumped nonlinear circuits have applied the harmonic input method, mentioned previously in section 3-2, to the nodal analysis. [13] introduces the application of harmonic input method to hybrid analysis. The latter approach is more general in that it allows for both current- and voltage-controlled nonlinear elements.

Our discussion in this section for solving multiple-node, multiple-nonlinearity circuits will be centered around the application of multi-dimensional transforms to a cutset type analysis. Thus, we will be solving for the generalized voltage ratio transfer functions. As we proceed with our discussion, it will become apparent that a cutset analysis approach is the most natural way of solving for the generalized voltage-ratio transfer functions. We now develop the procedure.

The first step in the analysis is to represent each nonlinear element by a polynomial expansion. Thus, in the distortion analysis of

transistor amplifiers [7,36], the exponential type controlled sources in the Ebers-Moll model are first represented by a Taylor series expansion of the function about the quiescent point, thereby yielding a polynomial in terms of the incremental variables. The types of nonlinear elements, and their series representation, that are commonly encountered are:

1. No memory, independent nonlinearity (Nonlinear Resistor)

$$i = F(v) = \sum_{j=1}^{\infty} a_j v^j \quad (3-56)$$

2. No memory, dependent nonlinearity

$$i = G(u, v) = \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} a_{jk} u^j v^k, \quad a_{00} = 0 \quad (3-57)$$

3. Capacitive, independent nonlinearity

$$i = \frac{d}{dt} Q(v) = \frac{d}{dt} \sum_{j=1}^{\infty} a_j v^j \quad (3-58)$$

4. Inductive, independent nonlinearity

$$i = \int_{-\infty}^t \phi(v) dt = \int_{-\infty}^t \sum_{j=1}^{\infty} a_j v^j dt \quad (3-59)$$

where

$i \equiv$  incremental current through the element

$v \equiv$  incremental controlling voltage

$u \equiv$  incremental controlling voltage

The general procedure employed to solve for the nonlinear transfer functions of a single-input, single-output nonlinear circuit using the cutset analysis approach is illustrated in Fig. 3-1 by considering each of the four nonlinear element types mentioned above.

Consider the nonlinear circuit  $N$ , shown in Fig. 3-1(a), containing a nonlinear resistor, a nonlinear dependent source, a nonlinear capacitor, and a nonlinear inductor, where each nonlinear element is voltage controlled. The procedure begins by identifying all the nonlinear elements, as shown in Fig. 3-1(b). We note that the four nonlinear elements depend on six voltages. The next step is to lump the linear parts of the nonlinear elements with the existing linear network to form the augmented linear network. The square, cubic, quartic, etc. terms of the nonlinearity are treated as nonlinear current sources, indicated by  $i_k^n$ , meaning the  $n$ th order current source at port  $k$ . Since the dependent source,  $g(v_5, v_6)$ , depends on voltages  $v_5$  and  $v_6$ , we also extract these as ports. Thus, altogether we end up with an 8-port linear network, as shown in Fig. 3-1(c).

The output variables to be found are the voltages at these eight ports. The augmented linear network is denoted by  $N'$  in Fig. 3-1(c). To solve for the voltage vector  $\underline{v} = [v_1 \ v_2 \ v_3 \ \dots \ v_8]$ , we immediately recognize that the branches across these voltage variables must be selected as part of the tree [29]. Clearly, some of the other branches in the augmented linear network may also appear as part of the tree. These will then appear as voltage variables in the cutset equations for the augmented linear network. Since there is no need for these addi-

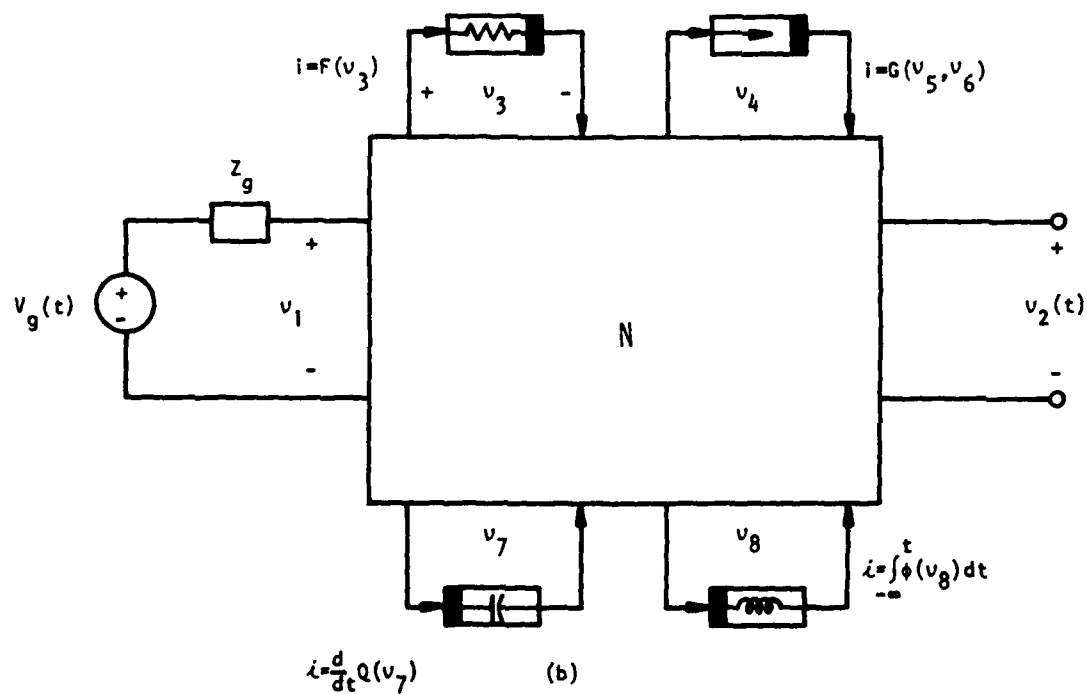
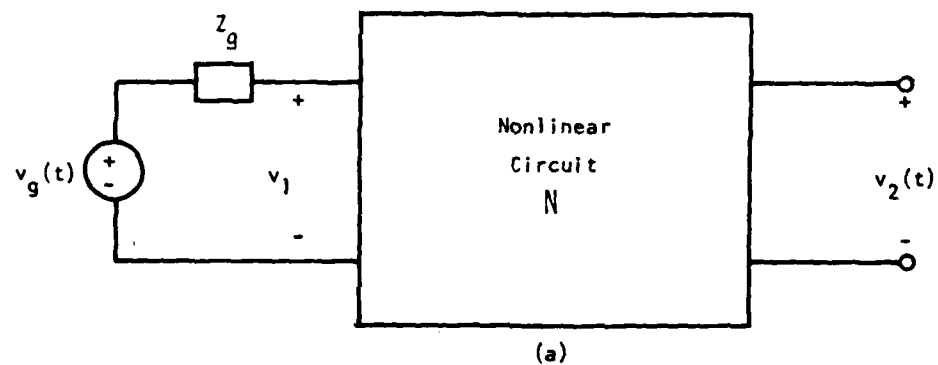


Figure 3-1. Steps in Nonlinear Circuit Analysis using Volterra Series.



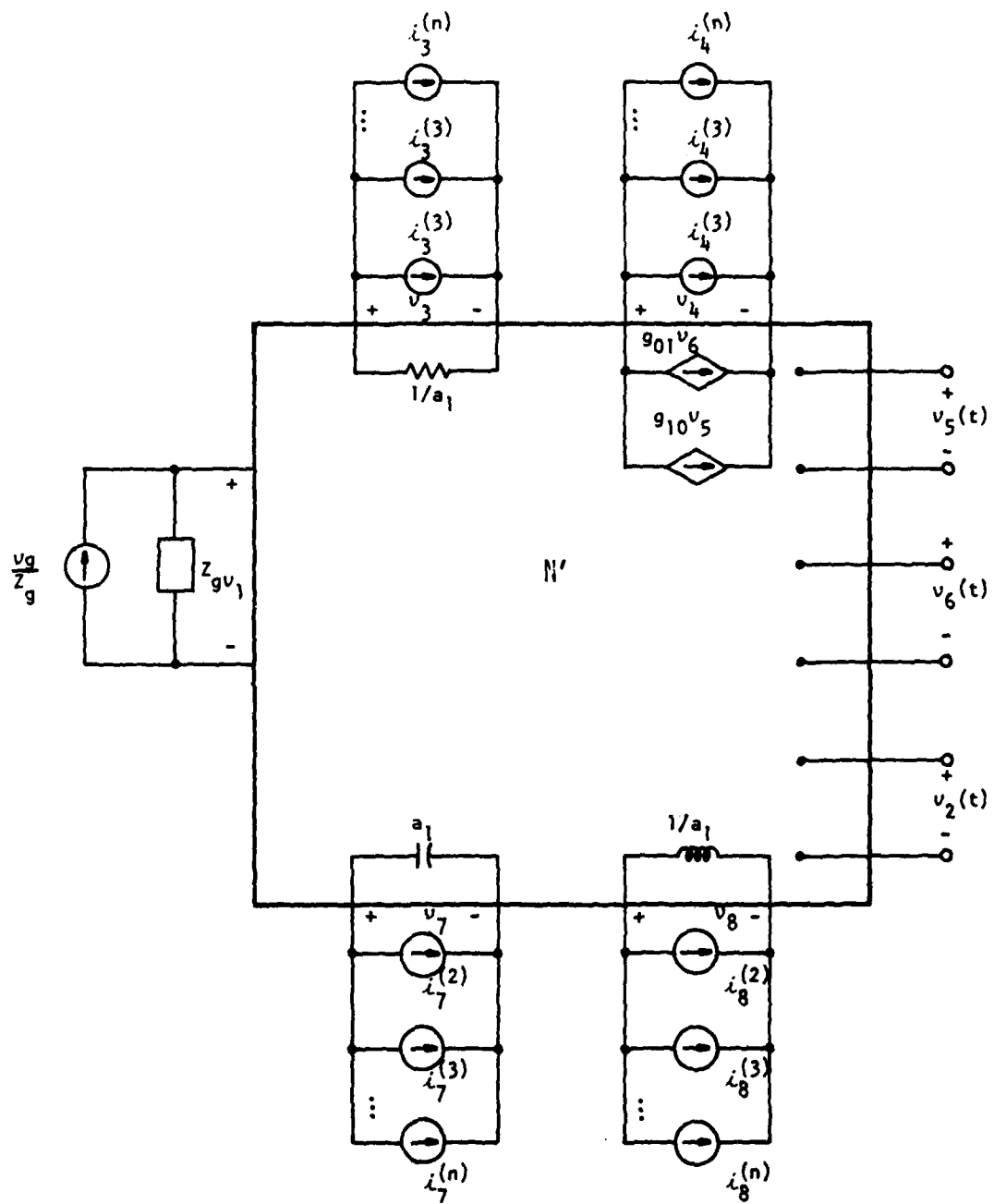


Figure 3-1. (contd.)

tional variables, we can reduce the dimensionality of our equations by a systematic elimination of these unwanted variables. In the case under consideration, we should be left with only the vector  $\underline{v} = [v_1 \ v_2 \ \dots \ v_8]$  as the unknown vector. Each of these 8 ports will have a set of transfer functions of order 1 to  $n$  associated with it. Our task here is to solve for these transfer functions.

At this point, we make the following general notational definitions:

$$\underline{H}_k(s_1, s_2, \dots, s_k) = \begin{bmatrix} H_k^1(s_1, \dots, s_k) \\ H_k^2(s_1, \dots, s_k) \\ \cdot \\ \cdot \\ H_k^m(s_1, s_2, \dots, s_k) \end{bmatrix} \quad (3-60)$$

where

$H_k^j \equiv$   $k$ th order nonlinear transfer function from the input to the  $j$ th port;  $m = 8$  in our example here.

$$\underline{v}(t) = [v_1(t) \ v_2(t) \ \dots \ v_m(t)]^T \quad (3-61)$$

where  $v_i \equiv$  voltage at the  $i$ th port

The cutset equations for the  $m$ -port nonlinear network can be written as:

$$\underline{Y}(p)\underline{v} + \underline{F}(\underline{v}) + \underline{G}(\underline{u}, \underline{v}) + p\underline{Q}(\underline{v}) + \frac{1}{p}\underline{\phi}(\underline{v}) =$$

$$[\underline{v}_g/z_g(p)][1 \ 0 \ 0 \ \dots \ 0]^T \quad (3-62)$$

where

$p \equiv$  differential operator,  $\frac{d}{dt}$

$\underline{Y}(p) \equiv$  Reduced admittance matrix for the p-port augmented linear network

$\underline{F}(\underline{v}) \equiv$  vector composed of all nonlinear currents through the zero memory independent nonlinearity

$\underline{G}(\underline{u}, \underline{v}) \equiv$  vector composed of all nonlinear currents through the zero memory dependent nonlinearities

$\underline{Q}(\underline{v}) \equiv$  vector composed of all nonlinear currents through the nonlinear capacitive nonlinearities

$\underline{\phi}(\underline{v}) \equiv$  vector composed of all nonlinear currents through the nonlinear inductive elements.

$z_g(p) \equiv$  source impedance

Since the linear parts of the functions  $\underline{F}(\cdot)$ ,  $\underline{G}(\cdot)$ ,  $\underline{Q}(\cdot)$ , and  $\underline{\phi}(\cdot)$  in eqn. (3-56) through (3-59) have been lumped together with the linear part of the network, the general form of these functions will be as follows:

$$\underline{Z}(\underline{v}) = \underline{Z}_2(\underline{v}) + \underline{Z}_3(\underline{v}) + \underline{Z}_4(\underline{v}) + \dots \quad (3-63)$$

where

$\underline{Z}_2(\underline{v})$  is a quadratic function of  $\underline{v}$

$\underline{Z}_3(\underline{v})$  is a cubic function of  $\underline{v}$

$\underline{Z}_4(\underline{v})$  is a quartic function of  $\underline{v}$

...

$\underline{Z}(\cdot)$  being  $\underline{F}(\cdot)$ ,  $\underline{G}(\cdot)$ ,  $\underline{Q}(\cdot)$ , or  $\phi(\cdot)$ . Thus, eqn. (3-62) can be re-written as:

$$\underline{Y}(p)\underline{v} = \frac{1}{\underline{z}_g(p)} \begin{bmatrix} \underline{v}_g(t) \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} - \underline{i}_k(t) \quad , k \geq 2 \quad (3-64)$$

where  $\underline{i}_k(t)$  denotes vectors of 2<sup>nd</sup> and higher order current sources due to  $\underline{F}(\underline{v})$ ,  $\underline{G}(\underline{u}, \underline{v})$ ,  $p\underline{Q}(\underline{v})$ , and  $\frac{1}{p}\underline{\psi}(\underline{v})$ . The mathematical artifice used in section (3-2) could have been applied here also to obtain the form of all the nonlinear current source terms,  $\underline{i}_k(t)$ . For the sake of brevity, we will not use that approach here, but simply use the results of section (3-2) to identify the different order current sources due to different nonlinearities. These are summarized in Table 3-1, where  $\underline{v}^i(t)$  denotes the  $i$ th order response voltage  $\underline{v}(t)$ , which control the nonlinear element characteristics.

Table 3-1. Nonlinear Current Sources in multiple-node, multiple-nonlinearity circuit analysis.

Nonlinear Resistor,  $F(v)$ :

$$k = 2: a_2[v^1]^2$$

$$k = 3: 2a_2[v^1 v^2] + a_3[v^1]^3$$

$$k = 4: a_2[2v^1 v^3 + (v^2)^2] + 3a_3[v^1]^2 v^2 + a_4[v^1]^4$$

Nonlinear Dependent Nonlinearity  $G(u, v)$ :

$$k = 2: a_{20}[u^1]^2 + a_{02}[v^1]^2 + a_{11}u^1 v^1$$

$$k = 3: a_{30}[u^1]^3 + a_{03}[v^1]^3 + a_{21}[u^1]^2 v^1 + a_{12}u^1 [v^1]^2 + 2a_{20}u^1 u^2 + 2a_{02}v^1 v^2 + a_{11}[u^1 v^2 + u^2 v^1]$$

$$k = 4: a_{40}[u^1]^4 + a_{04}[v^1]^4 + a_{13}u^1 [v^1]^3 + a_{22}[u^1]^2 [v^1]^2 + a_{21}(2u^1 u^3 + [u^2]^2) + a_{11}(u^3 v^1 + u^1 v^3 + u^2 v^2) + a_{02}(2v^1 v^3 + [v^2]^2) + 3a_{30}[u^1]^2 v^2 + 3a_{03}[v^1]^2 v^2 + a_{21}([u^1]^2 v^2 + 2u^1 u^2 v^1) + a_{12}(u^2 [v^1]^2 + 2u^1 v^1 v^2)$$

Nonlinear Capacitive Nonlinearity  $pQ(v)$ :

$$k = 2: a_2 p[v^1]^2$$

$$k = 3: 2a_2 p[v^1 v^2] + a_3 p[v^1]^3$$

$$k = 4: a_2 p(2v^1 v^3 + [v^2]^2) + 3a_3 p[v^1]^2 v^2 + a_4 p[v^1]^4$$

Table 3-1 (contd.)

Nonlinear Inductive Nonlinearity,  $[1/p]\phi(v)$

$$k = 2: \frac{a_2}{p} [v^1]^2$$

$$k = 3: \frac{2a_2}{p} [v^1 v^2] + \frac{a_3}{p} [v^1]^3$$

$$k = 4: \frac{a_2}{p} (2v^1 v^3 + [v^2]^2) + \frac{3a_3}{p} [v^1]^2 v^2 + \frac{a_4}{p} [v^1]^4$$

We observe that the nonlinear current source terms in Table 3-1 are similar to the nonlinear terms whose transforms were derived in section 3-2, except for the nonlinear dependent source terms, which are functions of two controlling voltages  $u$  and  $v$ . The form of the transforms of the nonlinear dependent source will, however, be similar to the other nonlinearity types. These can again be written by inspection. For example,

$$a_{20}[u^1(t)]^2 = a_{20}u^1(t_1)u^1(t_2) \leftrightarrow a_{20} \overline{U(s_1)U(s_2)} \quad (3-65)$$

$$a_{11}u^1(t)v^1(t) = a_{11}u^1(t_1)v^1(t_2) \leftrightarrow a_{11} \overline{U(s_1)V(s_2)} \quad (3-66)$$

$$a_{20}u^1(t)v^2(t) = a_{20}u^1(t_1)v^2(t_2, t_3) \leftrightarrow a_{20} \overline{U(s_1)V(s_2, s_3)} \quad (3-67)$$

and so on.

We also note that a  $k$ -th order current source term in Table 3-1 depends on responses of order less than  $k$ , which implies that, in order to calculate a transfer function of order  $k$ , we need to determine the transfer functions up to order  $(k-1)$ .

The first order transfer function can be solved for easily. It is simply the linear circuit response. Therefore,

$$\underline{Y}(p)\underline{v}(t) = \underline{i}_1(t) \quad (3-68)$$

For a single input system,  $\underline{i}_1(t) = 1/z_g [v_g(t) \ 0 \ 0 \ \dots \ 0]^T$ , where  $v_g(t)$  is the source voltage. Taking the transform of eqn. (3-68), and assuming that the input source to be an impulse function, we get:

$$\underline{v}^1(s_1) = \underline{H}_1(s_1) = 1/z_g [\underline{Y}(s_1)]^{-1} [1 \ 0 \ 0 \ \dots \ 0]^T \quad (3-69)$$

where  $\underline{H}_1(s_1)$  was defined in eqn. (3-60).

The equation for obtaining the second-order response, as per eqn. (3-64), is the following:

$$\underline{Y}(p)\underline{v}^{(2)}(t) = -\underline{i}_2(t) \quad (3-70)$$

Since the input to the nonlinear circuit is assumed to be an impulse function, the transform of eqn. (3-70), after using eqn. (3-14), is:

$$\underline{Y}(s_1+s_2)\underline{H}_2(s_1,s_2) = -\underline{I}_2(s_1,s_2) \quad (3-71)$$

The elements of vector  $\underline{I}_2(s_1,s_2)$  can be obtained by performing a two-dimensional transform on the terms associated with  $k = 2$  in Table 3-1. This operation, as indicated earlier, can be carried out by inspection. Thus, we have

$$\underline{H}_2(s_1,s_2) = -[\underline{Y}(s_1+s_2)]^{-1} \underline{I}_2(s_1,s_2) \quad (3-72)$$

Likewise we can solve for  $\underline{H}_3(s_1,s_2,s_3)$ . In general, we solve for the  $n$ th order transfer function using eqn. (3-73):

$$\underline{H}_n(s_1,s_2,\dots,s_n) = [\underline{Y}(\sum_{i=1}^n s_i)]^{-1} \underline{I}_n(s_1,\dots,s_n) \quad (3-73)$$

We observe a striking similarity between eqn. (3-73) and the equations for nodal or cutset analysis encountered in linear circuit analysis. A little thought would show that the process of solving eqn. (3-73) is identical to solving the linear circuit in Fig. 3-2. We have nonlinear current sources as inputs to the augmented linear circuit. A



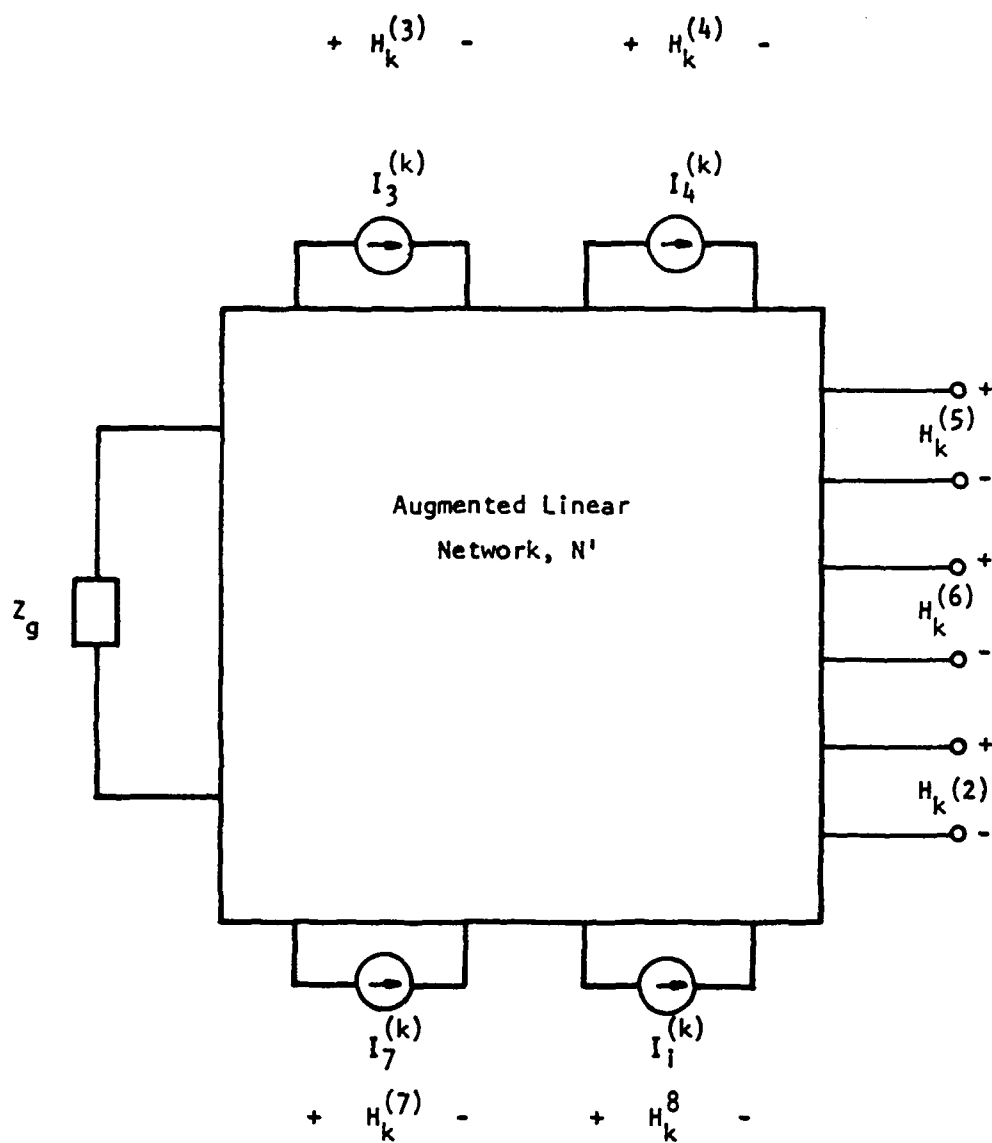


Figure 3-2. Determination of Volterra Transfer Functions.

k-th order vector of transfer functions is obtained by exciting the linear circuit by the kth order current sources. Just as in the case of linear systems, superposition can be applied here when a particular order response is determined from the lower order responses. That is, a k-th order response can be obtained by applying the k-th order current sources one-by-one at each of the ports and then summing up the responses. It is important to note, however, that the complete responses of order up to  $(k-1)$  must be determined before we can obtain the kth order response by superposition. It is also noted that the illustration of Fig. 3-2 is for pedagogic purpose and that the nonlinear current sources are not physically present in the circuit under consideration.

### 3-5. Multiple Input Circuit Analysis

Much of the foregoing discussion has been concerned with the analysis of nonlinear circuits with single inputs. However, many applications of practical significance in nonlinear circuit analysis have multiple inputs. For example, in a receiver system, the mixer circuit has two inputs: 1) the message signal, and 2) the local oscillator signal. The transmitter again has nonlinear circuits with multiple inputs. The Volterra series method is especially well suited for the analysis of such circuits. In this section we discuss how the various order transfer functions change as a result of multiple inputs.

From the discussion in section 3-4, it should be apparent that the analysis of nonlinear circuits using the Volterra series method involves the repeated analysis of a linearized circuit. The fundamental relationship had the following form (see eqn. 3-64):

$$\underline{Y}(p)\underline{V}(t) = \frac{1}{z_g(p)} \underline{i}_1(t) - \underline{i}_2(t) - \underline{i}_3(t) + \dots \quad (3-74)$$

where  $\underline{i}_k(t)$  is the  $k$ -th current source vector. The  $k$ -th order current source ( $k \geq 2$ ), as per our previous discussion, depends on up to the  $(k-1)$  order voltage ratio transfer functions, is injected at each of the ports at which the nonlinear elements are present, and is due entirely to the nonlinear characteristics of the nonlinearity. Furthermore, it is proportional to the  $k$  values of the circuit input multiplied together. Thus, the number of elements in the vector  $\underline{i}_k(t)$ ,  $k \geq 2$ , remain unchanged when multiple inputs are present; only the  $\underline{i}_1(t)$  vector is changed.

Consider, for example, the two-input circuit of Fig. 3-3(a). Then, to solve for the first-order transfer function, we write the vector transform equations as:

$$\underline{Y}(s_1)\underline{V}(s_1) = \underline{I}_1(s_1) \quad (3-75)$$

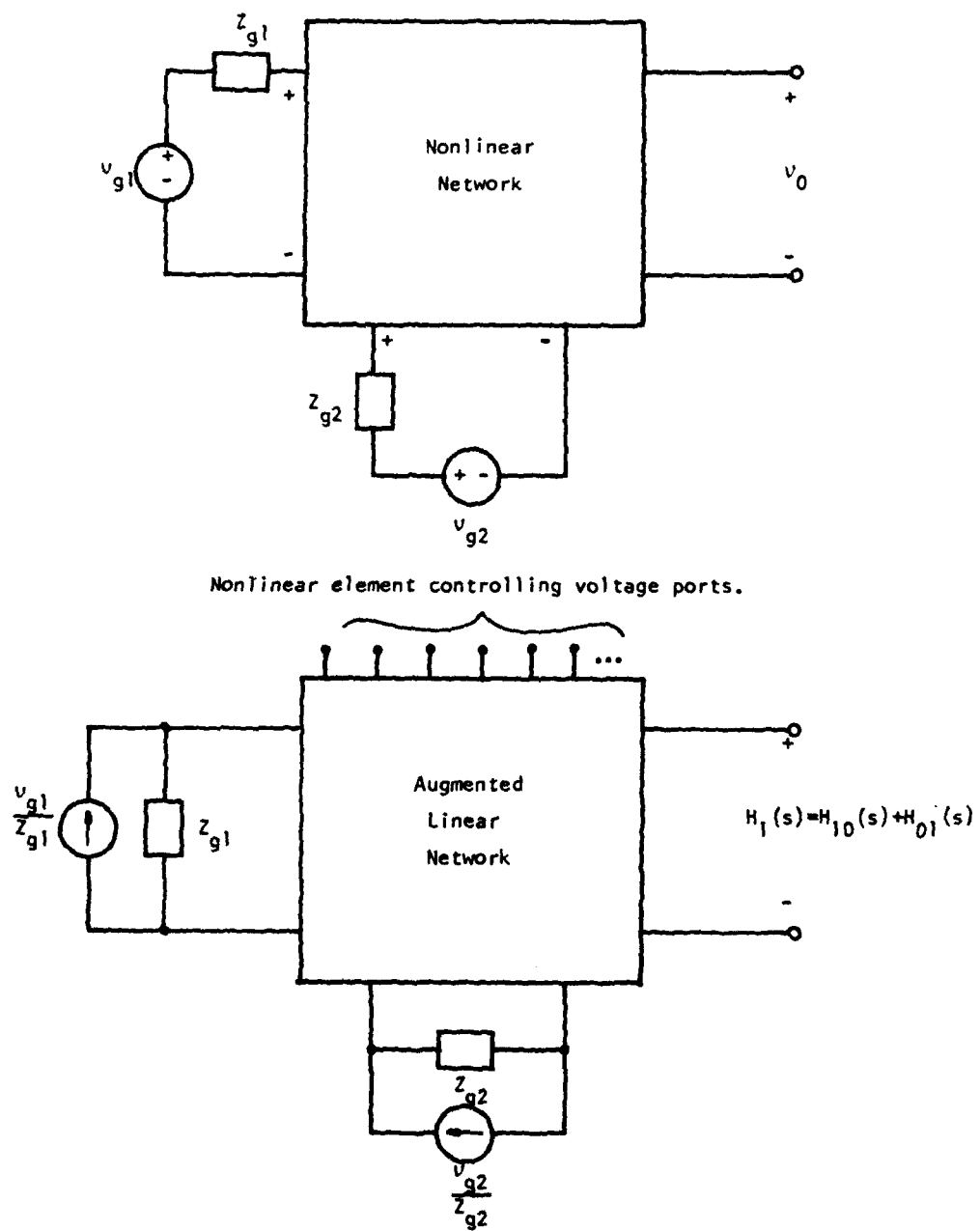
where

$$\underline{I}_1(s_1) = [Y_{g1}(s_1)V_{g1}(s_1) \quad Y_{g2}(s_1)V_{g2}(s_1) \quad 0 \quad \dots \quad 0]^T \quad (3-76)$$

and  $\underline{Y}$  and  $\underline{V}$  are as defined previously. The transfer function vector can be written as:

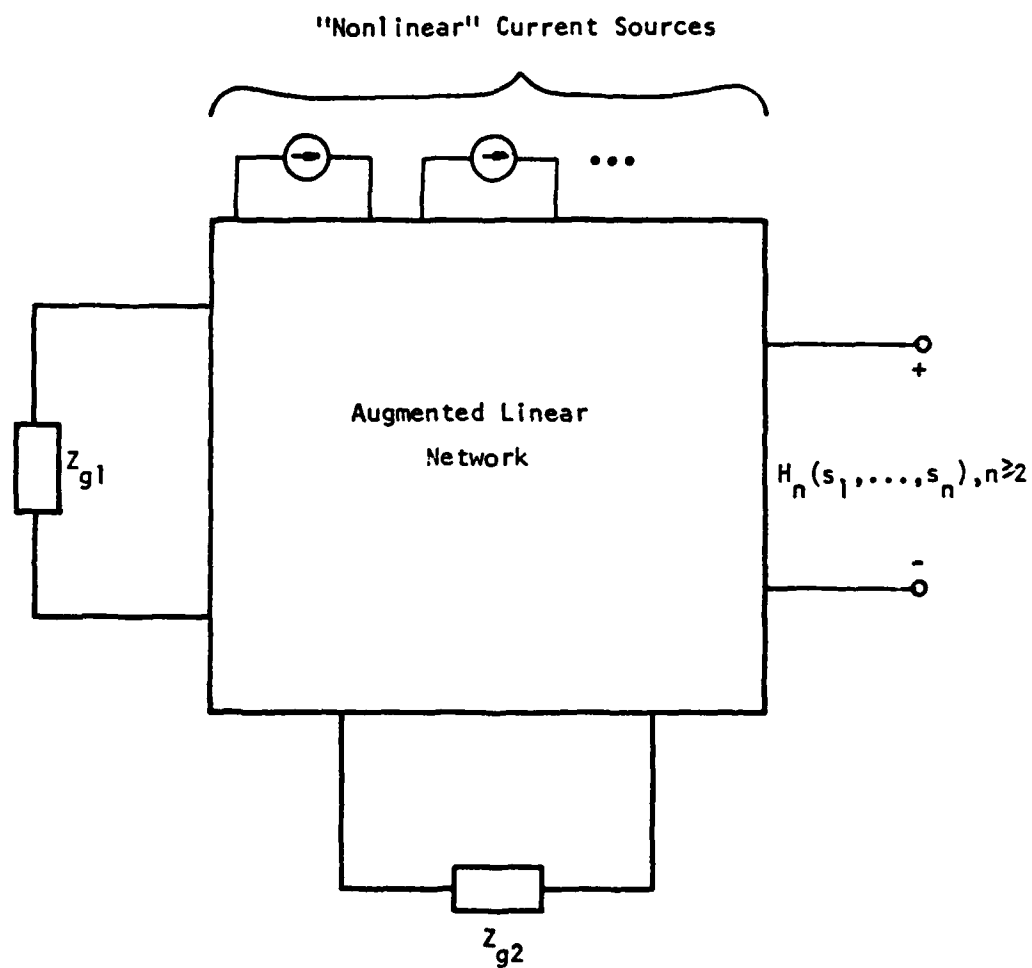
$$\underline{H}_1(s_1) = \underline{H}_{10}(s_1) + H_{01}(s_1) \quad (3-77)$$

where



(b) Circuit for determining first-order transfer function.

Fig. 3-3. Multiple Input Nonlinear Circuit Analysis.



(c) Circuit for determining  $H_n(s_1, s_2, \dots, s_n), n \geq 2$ .

Fig. 3-3. (contd.) Multiple Input Nonlinear Circuit Analysis.

$$\underline{H}_{10}(s_1) = \left[ \frac{V^{(1)}(s_1)}{V_{g1}} \quad \frac{V^{(2)}(s_1)}{V_{g1}} \quad \dots \quad \frac{V^{(p)}(s_1)}{V_{g1}} \right] T \bigg|_{V_{g2}=0} \quad (3-78)$$

and

$$\underline{H}_{01}(s_1) = \left[ \frac{V^{(1)}(s_1)}{V_{g2}} \quad \frac{V^{(2)}(s_1)}{V_{g2}} \quad \dots \quad \frac{V^{(p)}(s_1)}{V_{g1}} \right] T \bigg|_{V_{g1}=0} \quad (3-79)$$

where  $V^{(i)}$  is the voltage at port  $i$ .

The second- and higher-order transfer function vectors are solved for by removing the given input sources and applying the fictitious nonlinear current sources across the ports at which the nonlinear elements are present. The vector transform equation for solving for the second-order transfer function is still given by:

$$\underline{H}_2(s_1, s_2) = -[\underline{Y}(s_1 + s_2)]^{-1} [\underline{I}_2(s_1, s_2)] \quad (3-80)$$

where

$$\underline{I}_2(s_1, s_2) = [I^{(1)}(s_1, s_2) \quad I^{(2)}(s_1, s_2) \quad \dots \quad I^{(p)}(s_1, s_2)] \quad (3-81)$$

Depending on the nonlinearity type, the general form of  $I^{(l)}(s_1, s_2)$ , the second-order current source across port  $l$ , will be:

$$I_2^{(l)}(s_1, s_2) = a_2 H_1^{(l)}(s_1) H_1^{(l)}(s_2) \quad (3-82)$$

where  $H_1^{(l)}(*)$  is known from eqn. (3-77). The determination of the higher-order transfer functions is done similarly.

In summary, we note that the presence of multiple input sources in a nonlinear circuit does not drastically alter the procedure for deter-

mining the Volterra transfer functions. Only the structure of the first-order current source vector is changed as a result of multiple sources. This change is reflected in the values of the elements making up the second- and higher-order current source vectors, whose structure remains unchanged.

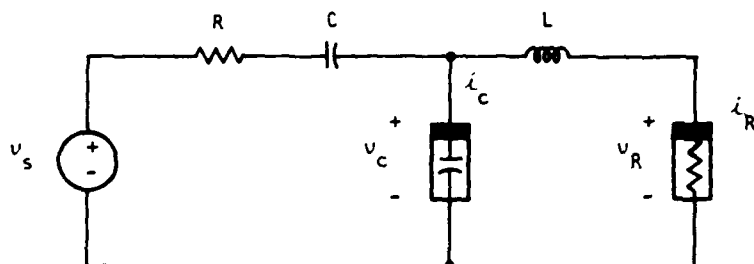
### 3-6. An Example

In this sub-section, we present an example to illustrate the ideas presented in the previous sub-sections. Specifically, we derive the nonlinear transfer functions for a multiple nonlinearity circuit.

Consider the circuit of fig. 3-4(a), which contains a resistive and a capacitive nonlinearity. The nonlinear element descriptions are also given in the Figure. The augmented linear network is shown in Fig. 3-4(b). The equilibrium equations for the nonlinear circuit can be written by inspection as:

$$\begin{bmatrix} G+pC & -pC & 0 \\ -pC & pC+pC_1+1/pL & -1/pL \\ 0 & -1/pL & 1/pL+g_1 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \begin{bmatrix} Gv_s(t) \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ C_2pv_2^2 + C_3pv_2^3 \\ g_2v_3^2 + g_3v_3^3 \end{bmatrix} \quad (3-83)$$

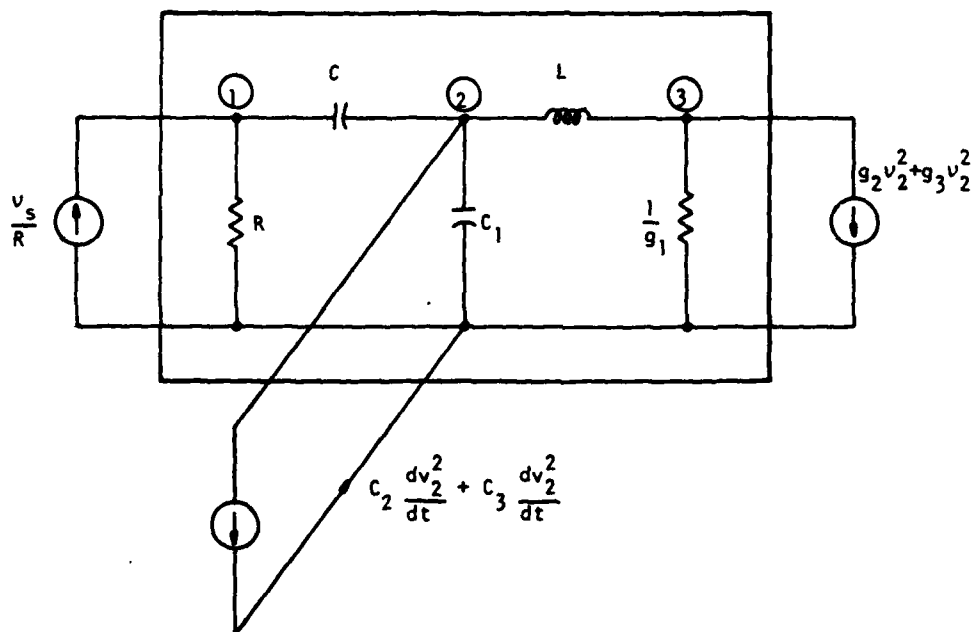
where  $v_i(t)$  is the  $i$ -th node voltage,  $p = \frac{d}{dt}$ , and  $G = 1/R$ . Each of the node voltages can be expanded into a Volterra series, thus giving



$$i_c = C_1 \frac{dv_c}{dt} + C_2 \frac{dv_c^2}{dt} + C_3 \frac{dv_c^3}{dt}$$

$$i_R = g_1 v_R + g_2 v_R^2 + g_3 v_R^3$$

(a)



(b)

Figure 3-4. A Nonlinear Circuit Example



$$v_i(t) = \sum_{k=1}^{\infty} v_i^{(k)}(t) \quad (3-84)$$

where  $v_i^{(k)}(t)$  represents a  $k$ -th order term in the solution for the  $i$ -th node voltage. Substituting eqn. (3-84) in eqn. (3-83) for  $i = 1, 2$ , and  $3$ , and equating terms of like order on both sides of eqn. (3-83), we get the following sets of equations for the various order responses.

First Order:

$$\begin{bmatrix} G+pC & -pC & 0 \\ -pC & pC+pC_1+1/pL & -1/pL \\ 0 & -1/pL & 1/pL+g_1 \end{bmatrix} \begin{bmatrix} v_1^{(1)}(t) \\ v_2^{(1)}(t) \\ v_3^{(1)}(t) \end{bmatrix} = \begin{bmatrix} Gv_s(t) \\ 0 \\ 0 \end{bmatrix} \quad (3-85)$$

Second Order:

$$\begin{bmatrix} G+pC & -pC & 0 \\ -pC & pC+pC_1+1/pL & -1/pL \\ 0 & -1/pL & 1/pL+g_1 \end{bmatrix} \begin{bmatrix} v_1^{(2)}(t) \\ v_2^{(2)}(t) \\ v_3^{(2)}(t) \end{bmatrix} = - \begin{bmatrix} 0 \\ C_2 p [v_2^{(1)}(t)]^2 \\ g_2 [v_3^{(1)}(t)]^2 \end{bmatrix} \quad (3-86)$$

Third Order:

$$\begin{bmatrix} G+pC & -pC & 0 \\ -pC & pC+pC_1+1/pL & -1/pL \\ 0 & -1/pL & 1/pL+g_1 \end{bmatrix} \begin{bmatrix} v_1^{(3)}(t) \\ v_2^{(3)}(t) \\ v_3^{(3)}(t) \end{bmatrix} =$$

$$- \begin{bmatrix} 0 \\ 2c_2 p[v_2^{(1)} v_2^{(2)}] + c_3 p[v_1^{(1)}]^3 \\ 2g_2 [v_3^{(1)} v_3^{(2)}] + g_3 [v_3^{(1)}]^3 \end{bmatrix} \quad (3-87)$$

The higher order response equations can be written similarly. We now solve for the transfer functions from the above equations.

Taking the one-dimensional transform of eqn. (3-85), we get:

$$\begin{bmatrix} G+s_1 C & -s_1 C & 0 \\ -s_1 C & s_1 (C+C_1)+1/s_1 L & -1/s_1 L \\ 0 & -1/s_1 L & 1/s_1 L+g_1 \end{bmatrix} \begin{bmatrix} v_1^{(1)}(s_1) \\ v_2^{(1)}(s_1) \\ v_3^{(1)}(s_1) \end{bmatrix} = \begin{bmatrix} G v_s(s_1) \\ 0 \\ 0 \end{bmatrix} \quad (3-88)$$

Assuming  $v_s(t) = \delta(t)$ , and therefore  $V_s(s_1) = 1$ , we get:

$$\underline{H}^{(1)}(s_1) = [\underline{Z}(s_1)][G \ 0 \ 0]^T \quad (3-89)$$

where

$$\underline{H}^{(1)}(s_1) = \begin{bmatrix} \frac{v_1^{(1)}(s_1)}{V_s} & \frac{v_2^{(1)}(s_1)}{V_s} & \frac{v_3^{(1)}(s_1)}{V_s} \end{bmatrix}^T \quad (3-90)$$

and

$$\underline{Z}(s_1) = \begin{bmatrix} G+s_1 C & -s_1 C & 0 \\ -s_1 C & s_1 (C+C_1)+1/s_1 L & -1/s_1 L \\ 0 & -1/s_1 L & 1/s_1 L+g_1 \end{bmatrix}^{-1} \quad (3-91)$$

To solve for the second-order transfer function vector,  $\underline{H}^{(2)}(s_1, s_2)$ , we must first recognize the following with ( $t_1 = t_2 = t$ ):

AD-A056 420

CLARKSON COLL OF TECHNOLOGY POTSDAM N Y  
LUMPED NONLINEAR SYSTEM ANALYSIS WITH VOLTERRA SERIES. (U)  
APR 80 H K THAPAR, B J LEON F30602-78

**F/G 9/3**

F30602-78-C-0102

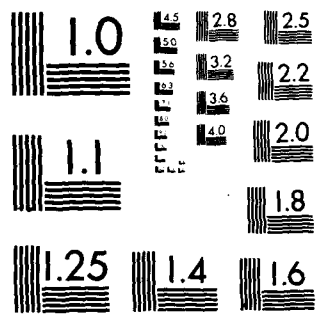
**UNCLASSIFIED**

**RADC-TR-80-140**

NL

$$\begin{aligned} \mathbb{Z} \text{ of } \mathbb{Z} \\ \text{Al.} \\ \Delta(\cdot) = \tau_{\Delta}(\cdot) \end{aligned}$$

END  
DATE  
FILMED  
8-80  
DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

$$\begin{aligned}
L[v_i^{(2)}(t)] &= L\left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_i^{(2)}(\tau_1, \tau_2) v_s(t_1 - \tau_1) v_s(t_2 - \tau_2) d\tau_1 d\tau_2\right] \\
&= H_i^{(2)}(s_1, s_2) V_s(s_1) V_s(s_2)
\end{aligned} \tag{3-92}$$

Since  $V_s(s_1) = V_s(s_2) = 1$ , the right-hand side of eqn. (3-92) is identically  $H_i^{(2)}(s_1, s_2)$ . Similarly, we write (with  $t_1 = t_2 = t$ ):

$$\begin{aligned}
L\left[[v_i^{(1)}(t)]^2\right] &= L\left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v_i^{(1)}(\tau_1) v_i^{(1)}(\tau_2) \delta(t_1 - \tau_1) \delta(t_2 - \tau_2) d\tau_1 d\tau_2\right] \\
&= v_i^{(1)}(s_1) v_i^{(2)}(s_2)
\end{aligned} \tag{3-93}$$

We note here that  $v_i^{(1)}(s_1) \equiv H_i^{(1)}(s_1)$  since  $v_s(t) = \delta(t)$ .

Taking the two-dimensional transform of eqn. (3-86), and substituting eqns. (3-92) and (3-93) in it, we get:

$$\begin{aligned}
\underline{H}^{(2)}(s_1, s_2) &= - \underline{Z}(s_1 + s_2) [0 \quad (s_1 + s_2) C_2 H_2^{(1)}(s_1) H_2^{(1)}(s_2) \\
&\quad g_2 H_3^{(1)}(s_1) H_3^{(1)}(s_2)]^T
\end{aligned} \tag{3-94}$$

where  $\underline{H}$ ,  $\underline{Z}$  are defined in eqns. (3-90) and (3-91).

To solve for the third-order transfer function, we proceed in a similar manner. However, as mentioned in section 3-4, symmetrization operation is used in solving for the third-order transfer function vector. The symmetrization operator is associated with the  $[v_i^{(1)}(t) v_i^{(2)}(t)]$  term. To see this, we write

$$[v_i^{(1)}(t)v_i^{(2)}(t)] = \frac{1}{3} [v_i^{(1)}(t_1)v_i^{(2)}(t_2, t_3) + v_i^{(1)}(t_2)v_i^{(2)}(t_1, t_3) + v_i^{(1)}(t_3)v_i^{(2)}(t_1, t_2)] \Big|_{t_1=t_2=t_3=t} \quad (3-95)$$

$$\equiv v_i^{(1)}(t_1)v_i^{(2)}(t_2, t_3) \quad (3-96)$$

We note that when we let  $t_1 = t_2 = t_3 = t$  in eqn. (3-95), the two sides are identically equal. As mentioned previously, the variables  $t_i$ 's are used for mathematical manipulations; the final analysis involves only one variable  $t$ . Thus, with this in mind, we note that the response, as a function of  $t$ , will remain unchanged when symmetrization is performed.

Proceeding in a manner similar to that for solving for the second-order transfer function vector, we obtain the following third-order transfer function vector:

$$\underline{H}^{(3)}(s_1, s_2, s_3) = \underline{Z}(s_1 + s_2 + s_3) \cdot$$

$$- \left[ \begin{array}{c} 0 \\ \frac{(s_1 + s_2 + s_3)[2C_2 H_2^{(1)}(s_1) H_2^{(2)}(s_2, s_3) + C_3 H_2^{(1)}(s_1) H_2^{(1)}(s_2) H_2^{(1)}(s_3)]}{2g_2 H_3^{(1)}(s_1) H_3^{(2)}(s_2, s_3) + g_3 H_3^{(1)}(s_1) H_3^{(1)}(s_2) H_3^{(1)}(s_3)} \end{array} \right] \quad (3-97)$$

We note that all the quantities on the right side of eqn. (3-97) are known, and thus we can solve for the third-order transfer functions. Our derivations here have been fairly detailed; it should, however, be

1

pointed out that the form of these transfer functions can be obtained in an algorithmic manner by inspection, thus rendering Volterra series method a viable approach for computer aided analysis of nonlinear circuits.

CHAPTER 4  
STEADY-STATE AND TRANSIENT ANALYSIS

4-1. Introduction

In the analysis of nonlinear systems, two main classes of solutions are generally sought: 1) steady state, and 2) transient response. In this section we show how these solutions can be obtained via the Volterra series method.

In this section we present how the steady state and the zero-state transient responses can be obtained from the generalized transfer functions, which can be obtained in an algorithmic manner - as was shown in section 3. For the sinusoidal steady-state response we deal with the case of multi-tone input; for the transient response, we treat the case of input signals which are Laplace transformable and factorable - which clearly includes a large class of signals used in everyday application. No numerical integration is required in obtaining both these solutions.

Section 4-2 deals with the subject of obtaining the sinusoidal steady-state solution using the generalized transfer functions. Section 4-3 presents the approach for obtaining the zero-state transient response.

4-2. Sinusoidal Steady-State Analysis

In linear system theory, the sinusoidal steady-state response is intimately tied to the transfer function of the system. A similar result is found for higher order responses using the Volterra series method: an  $n$ -th order response at a particular frequency is directly related to the  $n$ -th order transfer function. In this section we develop



this relationship.

If the harmonic input method [7,10-14] had been used in deriving the generalized transfer functions in section 3, the relationship between the n-th order steady state response and the n-th order transfer function would have been self-evident. But, since multi-dimensional transform theory was used to derive the generalized transfer functions, this relationship must be developed. We treat the specific case of  $n=2$  in section 4-2.1 and then derive the general relationship in section 4-2.2.

#### 4-2.1. Second-order Sinusoidal response:

The second-order output, according to the Volterra series, is given by:

$$y_2(t) = \int_0^{\infty} \int_0^{\infty} h_2(t-\tau_1, t-\tau_2) x(\tau_1) x(\tau_2) d\tau_1 d\tau_2 \quad (4-1)$$

Consider the input signal comprising two unit sinusoidal signals at frequencies  $\omega_a$  and  $\omega_b$ . The input  $x(\tau)$  is therefore:

$$x(\tau) = \left[ \frac{\exp(j\omega_a \tau) + \exp(-j\omega_a \tau)}{2} \right] + \left[ \frac{\exp(j\omega_b \tau) + \exp(-j\omega_b \tau)}{2} \right] \quad (4-2)$$

Substituting eqn. (4-2) in (4-1), we have:

$$y_2(t) = \int_0^{\infty} \int_0^{\infty} h_2(t-\tau_1, t-\tau_2) \cdot$$

$$\begin{aligned}
& \cdot \left[ \frac{\exp(j\omega_a \tau_1) + \exp(-j\omega_a \tau_1)}{2} + \frac{\exp(j\omega_b \tau_1) + \exp(-j\omega_b \tau_1)}{2} \right] \\
& \cdot \left[ \frac{\exp(j\omega_a \tau_2) + \exp(-j\omega_a \tau_2)}{2} + \frac{\exp(j\omega_b \tau_2) + \exp(-j\omega_b \tau_2)}{2} \right] \\
& \cdot d\tau_1 d\tau_2 \tag{4-3}
\end{aligned}$$

Considering one cross term only,

$$\int_0^\infty \int_0^\infty h_2(t-\tau_1, t-\tau_2) \frac{1}{4} \exp(j\omega_a \tau_1 + j\omega_b \tau_2) d\tau_1 d\tau_2 \tag{4-4}$$

and letting  $\sigma_1 = t - \tau_1$  and  $\sigma_2 = t - \tau_2$  and carrying out the integration yields,

$$\frac{1}{4} H_2(j\omega_a, j\omega_b) \exp[j(\omega_a + \omega_b)t] \tag{4-5}$$

Considering the other cross term similarly yields

$$\frac{1}{4} H_2(j\omega_b, j\omega_a) \exp[j(\omega_a + \omega_b)t] \tag{4-6}$$

However, if  $H_2(s_1, s_2)$  is symmetrical in its arguments, as they are assumed to be in this dissertation, then the terms in eqns. (4-5) and (4-6) are equal. The complex conjugate terms appear similarly. Hence, the output at frequency  $\omega_a + \omega_b$  is:

$$y(t)|_{\omega_a + \omega_b} = |H_2(j\omega_a, j\omega_b)| \cos[(\omega_a + \omega_b)t + \theta_{a+b}] \tag{4-7}$$

The  $2\omega_a$  or  $2\omega_b$  term and their complex conjugates appear only once in

eqn. (4-3); hence, their magnitude will be  $\frac{1}{2}|H_2(j\omega_a, j\omega_a)|$  and  $\frac{1}{2}|H_2(j\omega_b, j\omega_b)|$ , respectively. If only one frequency input was present, the results would be similar. The second-order output would then be:

$$y_2(t) = |H_2(j\omega_a, -j\omega_a)| + \frac{|H_2(j\omega_a, j\omega_a)|}{2} \cos(2\omega_a t + \theta_{2a}) \quad (4-8)$$

Thus, if we know  $H_2(s_1, s_2)$ , then the quantities in eqn. (4-8) can be easily evaluated. This is analogous to the case of linear systems, where the complex variable  $s$  is replaced by  $j\omega$  to compute the response at  $\omega$ .

If more than two-tones were present at the input, the second order response would be evaluated by taking all combinations of two frequencies at a time.

The response of the third and higher orders is similarly treated. We now present the general case.

#### 4-2.2. General Sinusoidal Steady-State Analysis.

In this sub-section, we develop the relationship which can be applied directly to compute the sinusoidal steady-state response of a nonlinear system from its nonlinear transfer functions, which can be obtained by the method presented in section 3. The discussion here relies heavily on [10].

Consider a nonlinear system excited by the sum of  $K$  distinct tones; i.e., defining  $N = 2K$ , we have,

$$x(t) = \frac{1}{2} \sum_{i=1}^N A_i \exp(j\omega_i t) \quad (4-9)$$

where  $\omega_i$  will include both positive and negative frequencies, and  $A_i$  for a negative frequency will be the complex conjugate of  $A_i$  for the positive frequency in order to have  $x(t)$  real. Then, the  $n$ th order output,  $y_n(t)$ , is given by:

$$y_n(t) = \int \cdots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i$$

$$= \int \cdots \int h_n(\tau_1, \dots, \tau_n) \frac{1}{2^n} \prod_{i=1}^n \sum_{k=1}^N A_k \exp[j\omega_k(t-\tau_i)] d\tau_i \quad (4-10)$$

Carrying out the product operation in eqn. (4-10), we get a function  $y_n(t)$  containing  $N^n$  terms, given by:

$$y_n(t) = \sum_{k_1=1}^N \cdots \sum_{k_n=1}^N \frac{1}{2^n} A_{k_1} \cdots A_{k_n} H_n(j\omega_{k_1}, \dots, j\omega_{k_n})$$

$$\cdot \exp[j(\omega_{k_1} + \dots + \omega_{k_n})t] \quad (4-11)$$

Notice that in arriving at eqn. (4-11), we have performed the  $\tau_i$  integration in eqn. (4-10), thus giving rise to the  $n$ -th order transfer function in eqn. (4-11). As the indices  $k_i$  are varied over the range 1 to  $N$ , many of the terms will be at the same frequency. The number of terms at various particular frequencies will vary according to what frequency combinations are taken. For example, in the case of  $n=2$  in section 4-2.1, there were two cross frequency terms, while there was only one second harmonic (at  $2\omega_a$ ) term. Similarly, for  $n=3$ , there are six terms in eqn. (4-11) at frequency  $\omega_a + \omega_b + \omega_c$ , three terms at  $2\omega_a + \omega_b$ , one

term at  $3\omega_a$ , etc. The nonlinear transfer functions, which make up the coefficients of these frequency terms, differ only in their arguments. However, since the transfer functions are assumed to be symmetric, the coefficient of the output at frequency  $\omega_a + \omega_b + \omega_c$  (in the case of  $n=3$ ) can be multiplied by 6. This obviates the need for taking all combinations to compute the output at  $\omega_a + \omega_b + \omega_c$ . Likewise we handle the case of other frequency combinations. With this insight, we can peek at the problem from a different perspective.

Let  $m_1, m_2, \dots, m_N$  be non-negative integers. Then, the number of terms at frequency  $\omega_\Sigma = m_1\omega_1 + m_2\omega_2 + \dots + m_N\omega_N$  is equal to the number of ways of forming  $m_1\omega_1 + \dots + m_N\omega_N$ . In the  $n$ -th order output spectrum to a multi-tone input, each term is evaluated by taking a distinct combination of  $n$  input tones at a time. To compute the  $n$ -th order output when the input frequencies are  $\omega_1, \omega_2, \dots, \omega_N$ , we must therefore restrict  $m_i$  in the following manner to compute  $\omega_\Sigma$ :

$$m_1 + m_2 + \dots + m_N = n \quad (4-12)$$

Now the problem reduces to the following: find the number of ways in which  $n$  objects can be divided into  $N$  groups of which the first contains  $m_1$  objects, the second  $m_2$  objects, etc. The solution to this problem is given by the multi-nomial coefficient [40]:

$$C_{n,N} = \frac{n!}{m_1! m_2! \dots m_N!} \quad (4-13)$$

By deriving eqn. (4-13), we have obviated the repetition of terms that is inherent in eqn. (4-11). An equivalent way of representing eqn. (4-11) through the use of eqn. (4-13) then becomes:

$$y_n(t) = \sum_{n,N} c_{n,N} \frac{A_1^{m_1} A_2^{m_2} \dots A_N^{m_N}}{2^n} \cdot H_n(j\omega_1, \dots, j\omega_2, \dots, \dots, j\omega_N, \dots) \cdot \exp[j(m_1\omega_1 + \dots + m_N\omega_N)t] \quad (4-14)$$

$\begin{matrix} m_1 \text{ times} & m_2 \text{ times} & & m_N \text{ times} \end{matrix}$

Since  $y_n(t)$  is real, eqn. (4-14) also contains the complex conjugate terms. Thus, the coefficient of the sinusoidal term at frequency  $m_1\omega_1 + \dots + m_N\omega_N$  in the  $n$ -th order output is given by:

$$c_{n,N} \frac{A_1^{m_1} A_2^{m_2} \dots A_N^{m_N}}{2^{n-1}} H_n(j\omega_1, \dots, j\omega_2, \dots, \dots, j\omega_N, \dots) \quad (4-15)$$

$\begin{matrix} m_1 \text{ times} & m_2 \text{ times} & & m_N \text{ times} \end{matrix}$

In computing the entire  $n$ -th order response in eqn. (4-14), we take all distinguishable combinations of  $m_i$  satisfying eq. (4-12). According to [40] there are

$$S_{n,N} = \binom{n+N-1}{n} = \frac{(n+N-1)!}{n!(N-1)!} \quad (4-16)$$

such combinations.

Equation (4-14) is the fundamental relationship between the  $n$ -th order output and the  $n$ -th order transfer function. At first glance, the evaluation of this equation appears to be a formidable task. But, after some thought, one finds that this is not such a difficult task after all. We, however, defer the discussion\* of this till section 5.

\*A computer implementation for  $n=2,3$  appears in [41]. One can extend it for any  $n$ , with the ultimate limitation being the storage.

We now illustrate the use of eqn. (4-15). We assume that the non-linear transfer functions are known. The case for  $n=2$  can be easily verified from the discussion in section 4-2.1. For a two-tone input at  $\omega_1$  and  $\omega_2$  and  $n=3$ , we have the following cases:

(a) The output at  $\omega_1$  and  $\omega_2$  have the following amplitudes, respectively:

$$y_3(t)|_{\omega_1} = \frac{3!|A_2|^2 A_1}{(4)1!1!1!} |H_3(j\omega_1, -j\omega_2, j\omega_2)| \quad (4-17)$$

$$y_3(t)|_{\omega_2} = \frac{3!A_2|A_1|^2}{(4)1!1!1!} |H_3(j\omega_1, -j\omega_1, j\omega_2)| \quad (4-18)$$

(b) The output at  $2\omega_1 + \omega_2$  has the following magnitude:

$$y_3(t)|_{2\omega_1 + \omega_2} = \frac{3!A_1^2 A_2}{(4)2!1!} |H_3(j\omega_1, j\omega_1, j\omega_2)| \quad (4-19)$$

(c) The output at  $3\omega_1$  has the following magnitude:

$$y_3(t)|_{3\omega_1} = \frac{3!(A_1)^3}{(4)3!} |H_3(j\omega_1, j\omega_1, j\omega_1)| \quad (4-20)$$

The other combinations can be carried out similarly. For the above cases we make the following observations: both eqns. (4-17) and (4-18) are similar to obtaining the output at  $\omega_a + \omega_b + \omega_c$ , and therefore we see a  $3! (=6)$  multiplication factor\*, which accounts for the six combinations at  $\omega_a + \omega_b + \omega_c$  that were mentioned earlier; eqn. (4-19) is similar to ob-

\*The constant factor 4 in the denominator appears consistently in all the output terms, and is therefore not regarded as a variable multiplication factor here. This factor appears due to the way  $x(t)$  was expressed in eqn. (4-9).

taining the output at  $2\omega_a + \omega_b$ , and therefore has a multiplication factor of 3, which again is in accordance with our earlier discussion; eqn. (4-20) is like evaluating the output at  $3\omega_a$ , and hence has a multiplication factor of 1.

In section (3-5), we dealt with the analysis of multiple input nonlinear circuits. In obtaining the sinusoidal steady-state response of such circuits the material of this section is still applicable. However, care must be taken in keeping track of the various input frequencies, and their associated transfer functions, when such an analysis is warranted.

#### 4-3. Transient Analysis using Volterra Series

In linear system analysis, the most fundamental relationship between the input and the output is the convolution integral. Thus, to compute the output response, one merely convolves the input function with the impulse response function. However, when the input function is Laplace transformable, the convolution operation is rarely performed and one resorts to the simpler transform methods for obtaining the system response. The transform method involves the use of the system transfer function.

A scheme, analogous to the transform domain method in linear system analysis, can be devised for obtaining the response of a nonlinear system via the Volterra series method. Again, the inputs allowable under this scheme must be Laplace transformable. This restriction is not very severe since most of the inputs considered in everyday applications are Laplace transformable. The scheme presented here uses the generalized transfer functions and the transform-domain description of the input to



compute the output. The response thus obtained is the zero-state time-response.

The basic problem considered here is: Given the generalized transfer functions of the nonlinear system and the transform of the input, determine the output response without performing a convolution in the transform domain. A procedure, due to George [5], helps to solve this problem by associating time variables in the transform domain. That is, given  $Y_2(s_1, s_2)$  as the transform of the output function  $y_2(t_1, t_2)$ , then  $Y_2(s)$ , the transform of  $y_2(t)$ , will be found directly from  $Y_2(s_1, s_2)$ . This technique is called "association of variables" and is applicable to the class of lumped systems. Lumped systems have the property that all the generalized transfer functions are factorable (just like the linear situation); that is, if

$$f(t_1, t_2) = f_1(t_1)f_1(t_2) \quad (4-21)$$

then

$$F(s_1, s_2) = F(s_1)F(s_2) \quad (4-22)$$

The examples in section 3 - in particular the example in section (3-6) - illustrate the concept of factorable transforms.

In the following sub-section we present the "association of variable" technique. In section 4-3.2 we apply this technique to the time-domain analysis of multiple node, multiple-nonlinearity circuits.

#### 4-3.1. Association of Transform Variables

The association of variables technique [5] is derived here for the second-order response. For higher-order transforms, one simply associates variables successively using the results for the second-order case.

The two-dimensional transform pair is related as follows:

$$Y_2(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y_2(t_1, t_2) \exp(-s_1 t_1 - s_2 t_2) dt_1 dt_2 \quad (4-23)$$

and

$$y_2(t_1, t_2) = \left(\frac{1}{2\pi j}\right)^2 \iint_{-\infty}^{\infty} Y_2(s_1, s_2) \exp(s_1 t_1 + s_2 t_2) ds_1 ds_2 \quad (4-24)$$

We would like to find

$$y_2(t_1) = y_2(t_1, t_1) \quad (4-25)$$

without actually performing the inverse transformation indicated by eqn. (4-24).

The transform of  $y_2(t_1)$  is  $Y_2(s_1)$ , where

$$Y_2(s_1) = \left(\frac{1}{2\pi j}\right) \int_{-\infty}^{\infty} y_2(t_1) e^{st_1} dt_1 \quad (4-26)$$

But from eqns. (4-24) and (4-25) we have

$$y_2(t_1) = y_2(t_1, t_1)$$

$$= \left(\frac{1}{2\pi j}\right)^2 \iint_{-\infty}^{\infty} ds_1 ds_2 Y_2(s_1, s_2) e^{(s_1 + s_2)t_1} \quad (4-27)$$

Letting  $s_1 + s_2 = s$ , we have

$$Y_2(t_1) = \frac{1}{2\pi j} \int_{-\infty}^{\infty} \exp(st_1) ds \cdot \left[ \frac{1}{2\pi j} \int_{-\infty}^{\infty} Y_2(s-s_2, s_2) ds_2 \right] \quad (4-28)$$

Equating (4-28) and (4-26), we have

$$Y_2(s) = \frac{1}{2\pi j} \int_{-\infty}^{\infty} Y_2(s-s_2, s_2) ds_2 \quad (4-29)$$

Equation (4-29) still involves a convolution operation, and hence is not too useful for our purpose. In the case of lumped systems, where the transfer functions are factorable, this convolution can be performed by inspection. For example, a typical second order transform is of the following form:

$$Y_2(s_1, s_2) = \frac{X}{s_1 + s_2 + x} \cdot \frac{Y}{s_1 + y} \cdot \frac{Z}{s_2 + z} \quad (4-30)$$

Then, substituting eqn. (4-30) in eqn. (4-29), we have:

$$Y_2(s) = \frac{1}{2\pi j} \int_{-\infty}^{\infty} \frac{X}{s-s_2+s_2+x} \cdot \frac{Y}{s-s_2+y} \cdot \frac{Z}{s_2+z} ds_2$$

$$= \frac{X}{s+x} \cdot \frac{1}{2\pi j} \int_{-\infty}^{\infty} \frac{Y}{s-s_2+y} \cdot \frac{Z}{s_2+z} ds_2 \quad (4-31)$$

The integral in eqn. (4-31) reduces to:

$$\frac{1}{2\pi j} \int_{-\infty}^{\infty} \frac{Y}{-s_2+(s+y)} \cdot \frac{Z}{s_2+z} ds_2 = \frac{YZ}{s+y+z} \quad (4-32)$$

Thus, we find that:

$$\frac{X}{s_1+s_2+x} \text{ becomes } \frac{X}{s+x}$$

and

$$\frac{Y}{s_1+y} \cdot \frac{Z}{s_2+z} \text{ becomes } \frac{YZ}{s+y+z}$$

A list of commonly encountered transform terms in lumped nonlinear systems along with their transforms after the association of variables is given in Table 4-1. A more detailed list can be found elsewhere [24].

Table 4-1. List of Associated Transforms

Multi-dimensional Transforms	Associated Transforms
$F(s_1+s_2+\dots+s_n)$	$F(s)$
$\frac{k}{(s_1+a)(s_2+b)}$	$\frac{k}{s+a+b}$
$\frac{k}{(s_1+s_2+a)(s_1+b)(s_2+c)}$	$\frac{k}{(s+a)(s+b+c)}$
$\frac{k}{(s_1+s_2+s_3+a)(s_1+b)(s_2+c)(s_3+d)}$	$\frac{k}{(s+a)(s+b+c+d)}$

Table 4-1. (contd.)

$$\frac{k}{(s_1+s_2+s_3+a)(s_2+s_3+b)(s_1+c)(s_2+c)(s_3+c)} \left( \frac{k}{2a-b} \right) \cdot \frac{1}{s+a} \left[ \frac{1}{s+b+c} - \frac{1}{s+3c} \right]$$

#### 4-3.2 Transient Analysis of Nonlinear Circuits

From the discussion in section 3, it is apparent that the analysis of lumped nonlinear circuits reduces to the repeated analysis of the augmented linear circuit excited by nonlinear current sources. The terms that make up the nonlinear current sources, as per Table 3-1, are factorable. Thus, the "association of variables" technique can be applied to obtain the transient analysis of nonlinear circuits. We now discuss how the various order responses can be obtained. We consider the unit step input case here; the cases for other factorable inputs can be handled similarly.

##### FIRST ORDER RESPONSE:

The general form of the first-order transfer function for the augmented linear network, ignoring multiple poles\*, is:

$$H_1(s_1) = \sum_{i=1}^N \frac{P_i}{s_1 + p_i} + \sum_{k=0}^M R_k s_1^k \quad (4-33)$$

where  $P_i$ ,  $p_i$  and  $R_k$  are complex constants. Then if  $X(s_1)$ , the transform of input  $x(t)$ , is factorable, the response  $Y_1(s)$  will have the same form as eqn. (4-33). We note here most of the inputs used in practice are factorable; that is, they can be expanded into partial expansion form as

---

\*The case of multiple poles can also be handled by the "association of variables" technique. In our discussion here, we do not treat this case.

in eqn. (4-33). Specifically, if  $x(t)$  is a step function, then the output,  $Y_1(s_1)$ , will have the form:

$$Y_1(s_1) = \frac{A_0}{s_1} + \sum_{i=1}^N \frac{A_i}{s_1 + p_i} + \sum_{k=0}^{M-1} a_k s_1^k \quad (4-34)$$

The time-domain response can be easily determined from eqn. (4-34) by taking the inverse Laplace transform.

#### SECOND ORDER RESPONSE:

The equation for determining the second-order transfer function is (see section 3.3):

$$\underline{H}_2(s_1, s_2) = -\underline{Z}(s_1 + s_2) \underline{I}_2(s_1, s_2) \quad (4-35)$$

where  $\underline{H}_2$  is the vector of second-order transfer functions for the multiple node, multiple nonlinear circuit,  $\underline{Z}(s_1 + s_2)$  is the inverse of the reduced admittance matrix, and  $\underline{I}_2(s_1, s_2)$  is the vector of second-order current sources. For the four types of nonlinear elements described in section 3-3, the elements of the vector  $\underline{I}_2(s_1, s_2)$  will be of the following form, depending on the type of nonlinear element:

#### Nonlinear Resistor:

$$I_2^D(s_1, s_2) = a_2 H_1^D(s_1) H_1^D(s_2) \quad (4-36)$$

#### Nonlinear Dependent Source:

$$I_2^D(s_1, s_2) = a_{20} H_1^r(s_1) H_1^r(s_2) + a_{02} H_1^q(s_1) H_1^q(s_2)$$

$$+ \frac{1}{2} a_{11} [H_1^r(s_1) H_1^q(s_2) + H_1^q(s_1) H_1^r(s_2)] \quad (4-37)$$

Nonlinear Capacitor:

$$I_2^p(s_1, s_2) = (s_1 + s_2) a_2 H_1^p(s_1) H_1^p(s_2) \quad (4-38)$$

Nonlinear Inductor:

$$I_2^p(s_1, s_2) = a_2 H_1^p(s_1) H_1^p(s_2) / (s_1 + s_2) \quad (4-39)$$

where  $I_2^p$  is the current source at port p,  $a_i$  is the coefficient of the quadratic term of the nonlinearity, and  $H_1^j$  is the first-order transfer function from the input port to port j. Thus, depending on the number of nonlinear elements, the second-order transfer function  $H_2^l$  can be expressed using eqn. (4-35) as:

$$H_2^l(s_1, s_2) = \sum_{k=1}^K z_{lk}(s_1 + s_2) I_2^k(s_1, s_2) \quad (4-40)$$

where  $z_{lk}(s_1 + s_2)$  is the l,k element of  $Z(s_1 + s_2)$  and K is the number of second-order current sources.

Observing the form of eqns. (4-36) through (4-39), we recognize that the output transfer function will be made up of a summation of terms of the following form:

$$\hat{H}_2(s_1, s_2) = 2(s_1 + s_2) H_1(s_1) H_1(s_2) \quad (4-41)$$

where  $\hat{H}_2$  denotes the "partial" transfer function, and 2 is obtained after the  $(s_1 + s_2)$  term appearing in eqn. (4-38) and (4-39) has been lumped with  $z(s_1 + s_2)$ . Then for an input  $x(t)$  having transform  $X(s)$ , we

have, the partial output,  $\hat{Y}_2(s_1, s_2)$ , given by:

$$\begin{aligned}\hat{Y}_2(s_1, s_2) &= \hat{H}_2(s_1, s_2) X(s_1) X(s_2) \\ &= 2(s_1 + s_2) H_1(s_1) X(s_1) H_1(s_2) X(s_2)\end{aligned}\quad (4-42)$$

Recalling that  $H_1(s_1) X(s_1)$  is the linear response at a particular port in the circuit, we can re-write eqn. (4-42) as:

$$\begin{aligned}\hat{Y}_2(s_1, s_2) &= 2(s_1 + s_2) Y_1(s_1) Y_1(s_2) \\ &= 2(s_1 + s_2) \sum_{i,j} \frac{A_i}{s_1 + p_i} \frac{A_j}{s_2 + p_j}\end{aligned}\quad (4-44)$$

Equation (4-44) has the same form as eqn. (4-30). Therefore, we can apply the "association of variables" technique by inspection to solve for  $\hat{Y}_2(s)$ , and then for  $\hat{y}_2(t)$ . By taking all the terms in eqn. (4-40), we can obtain  $y_2(t)$ , the second order time-domain response to an input  $x(t)$ . It should be noted that the terms  $\sum_k a_k s^k$ , which were present in eqn. (4-34), have been dropped from eqn. (4-44). This is because functions such as impulses, doublets, etc. do not exist when squared.

### THIRD ORDER RESPONSE:

The equation for determining the third-order transfer function vector,  $\underline{H}_3(s_1, s_2, s_3)$ , is:

$$\underline{H}_3(s_1, s_2, s_3) = -\underline{Z}(s_1 + s_2 + s_3) \underline{I}_3(s_1, s_2, s_3) \quad (4-45)$$

where  $\underline{Z}(\cdot)$  is the inverse of the reduced admittance matrix and  $\underline{I}_3(\cdot)$  is the third-order current source vector. The elements of the third-order



current source vector depends on the type of the nonlinearity. The form of these elements are as follows:

Nonlinear Resistor:

$$I_3^P(s_1, s_2, s_3) = 2a_2 H_1^P(s_1) H_2^P(s_2, s_3) + a_3 H_1^P(s_1) H_1^P(s_2) H_1^P(s_3) \quad (4-46)$$

Nonlinear Dependent Source:

$$\begin{aligned} I_3^P(s_1, s_2, s_3) = & 2a_{20} H_2^r(s_1, s_2) H_1^r(s_3) + 2a_{02} H_1^q(s_1) H_2^q(s_2, s_3) \\ & + a_{11} [H_1^r(s_1) H_2^q(s_2, s_3) + H_2^r(s_1, s_2) H_2^q(s_3)] \\ & + a_{30} \prod_{i=1}^3 H_1^r(s_i) + a_{03} \prod_{i=1}^3 H_1^q(s_i) \end{aligned} \quad (4-47)$$

Nonlinear Capacitor:

$$\begin{aligned} I_3^P(s_1, s_2, s_3) = & (s_1 + s_2 + s_3) [2a_2 H_1^P(s_1) H_2^P(s_2, s_3) \\ & + a_3 H_1^P(s_1) H_1^P(s_2) H_1^P(s_3)] \end{aligned} \quad (4-48)$$

Nonlinear Inductor:

$$I_3^P(s_1, s_2, s_3) = [2a_2 H_1^P(s_1) H_2^P(s_2, s_3)$$

$$+ a_3 H_1^p(s_1) H_i^p(s_2) H_1^p(s_3) I_3^p(s_1+s_2+s_3) \quad (4-49)$$

where  $I_3^p(\cdot)$  is the current source at port  $p$ ,  $a_i$  is the coefficient of the nonlinearity at port  $p$ , and  $H_1^i$  and  $H_2^i$  are the first- and second-order transfer functions at port  $i$ . Using eqn. (4-45), the third-order transfer function at port  $l$  can be expressed in terms of the individual third-order current sources in the network as follows:

$$H_3^l(s_1, s_2, s_3) = \sum_{k=1}^K z_{lk}(s_1+s_2+s_3) I_3^k(s_1, s_2, s_3) \quad (4-50)$$

where  $z_{lk}$  is the  $l, k$ -th element of matrix  $\underline{z}$  and  $K$  is the number of third-order current sources.

Observing the basic form of eqns. (4-46) through (4-49), we see a striking similarity. There are basically two types of terms making up the third-order current sources: 1) due to the squaring operating, giving rise to  $H_1(s_1) H_2(s_2, s_3)$ ; and 2) due to the cubing operation giving rise to  $\prod_{i=1}^3 H(s_i)$ . The transfer function at port  $l$ , from eqn. (4-50), is a summation, a typical term of which will have the following form:

$$\begin{aligned} \bar{A}_3(s_1, s_2, s_3) &= 2(s_1+s_2+s_3) H_1(s_1) H_2(s_2, s_3) \\ &+ 2(s_1+s_2+s_3) H_1(s_1) H_1(s_2) H_1(s_3) \end{aligned} \quad (4-51)$$

where  $\bar{A}_3$  is the "partial" third-order transfer function, and  $2$  is obtained after the  $(s_1+s_2+s_3)$  term appearing in eqn. (4-48) and (4-49) has been lumped with  $z(s_1+s_2+s_3)$ .

For an input  $x(t)$  having a transform  $X(s)$ , the partial response,  $\bar{Y}(s_1, s_2, s_3)$ , is given by:

$$\begin{aligned}
\hat{Y}_3(s_1, s_2, s_3) &= \hat{A}_3(s_1, s_2, s_3) X(s_1) X(s_2) X(s_3) \\
&= [2(s_1 + s_2 + s_3) H_1(s_1) H_2(s_2, s_3) \\
&\quad + 2(s_1 + s_2 + s_3) H_1(s_1) H_1(s_2) H_1(s_3)] \prod_{i=1}^3 X(s_i) \quad (4-52)
\end{aligned}$$

We now seek to obtain  $\hat{Y}_3(s)$ , the transform of  $\hat{Y}_3(t)$ , from eqn. (4-52). To do this, we look at each term in eqn. (4-52) separately.

Recalling\* that  $H_2(s_1, s_2) = \Sigma 2(s_1 + s_2) H_1(s_1) H_1(s_2)$ , we can re-write the first term of eqn. (4-52) after performing the symmetrization operation as:

$$\begin{aligned}
&2(s_1 + s_2 + s_3) 2(s_2 + s_3) H_1(s_1) H_1(s_2) H_1(s_3) X(s_1) X(s_2) X(s_3) \\
&= \frac{1}{3} 2(s_1 + s_2 + s_3) [2(s_1 + s_2) + 2(s_1 + s_3) + 2(s_2 + s_3)] \\
&\quad \prod_{i=1}^3 H_1(s_i) X(s_i) \quad (4-53)
\end{aligned}$$

Equation (4-53) is in a form suitable for applying the "association of variables" technique. We associate two variables at a time to reduce  $\hat{Y}_3(s_1, s_2, s_3)$  to  $\hat{Y}_3(s)$ . For example, consider the following association steps:

\*see eqns. (4-40) and (4-41).

$$\begin{aligned}
& 2(s_1+s_2+s_3)2(s_1+s_2)\prod_{i=1}^3 H_1(s_i)X(s_i) \\
& = 2(s_1+s_2+s_3)2(s_1+s_2)Y_1(s_1)Y_1(s_2)Y_1(s_3) \Big|_{\partial s_1, s_2} \\
& = 2(s_1+s_3)Q(s_1)Y_1(s_3) \Big|_{\partial s_1, s_3}
\end{aligned}$$

$$= R(s) \quad (4-54)$$

The association of two variables is identical to the association involved in determining the second order response. Thus, the partial third-order response is obtained by repeating this step once.

The association of variables involved in the second term in eqn. (4-51) is quite straightforward. Recalling that  $H_1(s_i)X(s_i) = Y_1(s_i)$  we can write the second term in eqn. (4-51) as:

$$2(s_1+s_2+s_3)\prod_{i=1}^3 Y_1(s_i) = 2(s_1+s_2+s_3)\prod_{i=1}^3 \left( \frac{A_j}{s_i+p_j} \right) \quad (4-55)$$

A typical term in eqn. (4-55) will be of the following form:

$$P(s_1, s_2, s_3) = 2(s_1+s_2+s_3) \frac{A_1}{s_1+p_1} \frac{A_2}{s_2+p_2} \frac{A_3}{s_3+p_3} \quad (4-56)$$

Associating two variables at a time, the final form,  $P(s)$ , is given by:

$$P(s) = 2(s) \frac{A_1 A_2 A_3}{s + p_1 + p_2 + p_3} \quad (4-57)$$

In our above discussion we isolated the typical terms present in the higher order responses to illustrate the application of the association of variables techniques to reduce these terms into a form suitable for obtaining the transient response. By summing up the partial responses, we can obtain the complete response for the network.

The association of variables involved in determining the complete response can be done by inspection, provided we have all information regarding the poles, and the associated residues, of the linearized system. For adapting this scheme on the computer, an algorithm which provides all this information must be used. A semi-symbolic analysis [29] of the linearized system must therefore be part of the algorithm for adapting Volterra series computer aided analysis. The main problem in implementing this scheme on the computer is the need to repeatedly perform a partial fraction expansion. For nonlinear resistive networks, this task is not so difficult if we make use of some of the basic properties of lumped linear systems. We will, however, defer this topic until chapter 5.

Finally, it is noted that the zero-input time domain response can be obtained by using the ideas presented above. For a circuit with multiple dynamic elements, this reduces to the multiple-input, multiple-output problem, where each input corresponds to the initial condition across the energy storage element.

CHAPTER 5  
COMPUTER-AIDED ANALYSIS USING VOLTERRA SERIES

5-1. Introduction

The adapting of Volterra series method in a general simulation program has been regarded as difficult by various authors [16,29,43]. As such, virtually no effort has been spent on investigating the computational aspect of this method and using it for the spectrum and time-domain analysis of general nonlinear circuits with polynomial type nonlinearities. Previous works [7,36] have endeavored to check the validity of this approach by applying it to specific circuit problems using a computer, but have never implemented the approach in a general simulation program.

The only major effort in using Volterra series for general nonlinear circuit analysis has been the development of the program NCAP [10,44]. A cursory review of this program reveals the inherent inefficiency in the computational approach with regards to storage and types of algorithms used. This inefficiency notwithstanding, there are severe limitations regarding the usefulness of the program: first, the program merely computes the numerical values of the nonlinear transfer function at the various program-prescribed combinations of the input frequencies, and does not compute all the transfer function values which are required to compute the complete output spectrum. Thus, NCAP does not yield the entire output spectrum information. Second, to compute up to an  $n$ -th order transfer function, the user must specify  $n$  input frequencies,

which are assumed to be a sum of exponentials and not real sinusoids. The program, therefore, is severely limited in its usefulness from the point of view of a user, who may only be interested in obtaining the output spectrum - say, for example, up to the third order response to two sinusoidal inputs - and has little use for the numerical values of the transfer functions at the program prescribed frequencies.

The zero-state time-domain analysis of nonlinear circuits using Volterra series approach on a computer has not been attempted before - not even for a specific nonlinear circuit problem.

In this section we look at the computational aspect of the Volterra series method for general simulation purposes and then present the basic algorithms for adapting this method for 1) spectrum and distortion analysis, and 2) for zero-state time-domain analysis to a step input. A digital computer program, PRANC (Program for Analysing Nonlinear Circuits), which makes use of the algorithms, has been written and implemented on the CDC 6500 computer. A detailed description and program listing is contained in a separate technical report [41]. Some examples from the use of this program are given in section 6.

In section 5-2, we present a brief overview of symbolic analysis in linear circuits, and then describe the reason why a symbolic approach is particularly useful in adapting Volterra series for general simulation. Section 5-3 deals with the implementation of the symbolic approach, and also contrasts the computational effort between a numerical approach and the particular symbolic approach used here. The algorithm for obtaining the complete output spectrum and the various distortion indices is described in section 5-4; the algorithm for zero-state time-domain

analysis to a step input is presented in section 5-5. A description of the computer implementation of these algorithms is given in section 5-6.

#### 5-2. Why a Symbolic Analysis Approach.

The symbolic analysis of circuits involves the computation of the  $a_i$  and  $b_i$  for network functions in the form

$$F(s) = \frac{N(s)}{D(s)} = \frac{\sum a_i s^i}{\sum b_i s^i} \quad (5-1)$$

when all circuit elements are known. The more general form

$$F(s; x_1, x_2, \dots, x_n) = \frac{N(s; x_1, \dots, x_n)}{D(s; x_1, \dots, x_n)} \quad (5-2)$$

applies when some elements of the circuit  $x_i$  are kept as symbols. The advantages of symbolic analysis have been recognized previously [29,45]. One particular advantage, and the one which is relevant to our problem here, is that the numerical evaluation of a function at discrete points is much easier and faster once the symbolic function is obtained than working repeatedly with a circuit analysis program. With this brief overview of symbolic analysis, we now proceed to answer the question: Why use a symbolic analysis approach for adapting the Volterra series method for general circuit analysis?

As pointed out in the previous sections, a nonlinear circuit is completely characterized by its Volterra kernels, or their transforms - the generalized transfer functions. These transfer functions are then directly related to the various order sinusoidal steady-state responses, as described in Chapter 4. The  $n$ -th order transfer function is deter-



mined from the following equation (see Chapter 3):

$$\underline{H}_n(s_1, \dots, s_n) = [\underline{Y}(\sum_{i=1}^n s_i)]^{-1} \underline{I}_n(s_1, \dots, s_n) \quad (5-3)$$

where  $\underline{Y}(\sum_{i=1}^n s_i)$  is the reduced node admittance matrix evaluated as  $s_1 + s_2 + \dots + s_n$ , and  $\underline{I}_n$  is the  $n$ -th order current source vector due to the nonlinear elements. To compute the output spectrum, we evaluate  $\underline{H}_n$  at various and many frequency combinations. From eqn. (5-3) it should be clear that such an evaluation will entail the inversion of the reduced node admittance matrix at each of these frequency combinations. Using combinational analysis, it has been shown [40] that for an input consisting of  $M$  sine waves, the number of inversions involved in an  $n$ -th order response, given by  $N_{n,m}$ , is:

$$N_{n,m} = \binom{2M+n-1}{n} \quad (5-4)$$

Thus, for a 5-tone input and up to a third order analysis, the number of inversions is approximately 285. For higher order responses, this number grows up very rapidly.

There are two basic approaches available to handle this inversion process: 1. Numerical approach, or 2. Symbolic approach. The advantage of evaluating symbolic transfer functions mentioned earlier makes the symbolic approach more attractive. How much advantage is gained in using a symbolic analysis depends on how much computational effort is expended in obtaining the symbolic inverse of the reduced node admittance matrix; thus, we need an efficient scheme for obtaining the symbolic inverse. The determination of the symbolic inverse will be the

subject of section 5-3.

Another reason for using a symbolic analysis is concerned with the determination of the zero-state time-domain response. The basic step involved in the "association of variables" technique, which was presented in section 4-3, is the prescribed combining of the various poles, and their associated residues, of the augmented linear network. A pole location and its residue can be readily determined once the transfer function, as per eqn. (5-1), is known. The entries of the symbolic inverse of the reduced node admittance matrix, which form the higher order transfer functions, will contain all the information needed to perform the prescribed combining of the various poles and residues to determine the zero-state response.

The reasons presented above stem from looking at the computational aspect of adapting Volterra series for computer-aided analysis. There are other advantages gained from using a symbolic analysis. An important one is that the generalized transfer functions can be obtained as functions of  $s_i$  once the inverse of the reduced node admittance matrix is obtained as a symbolic function of  $s$ . This can be seen from examining eqn. (5-3). The formation of the  $n$ -th order current source vector is a bootstrapping operation, as was pointed out in Chapter 3. That is, an  $n$ -th order source is formed from transfer functions of order less than  $n$ . The first-order transfer function vector is determined from a column\* of the symbolic inverse of the reduced node admittance matrix. The second order current sources, which depend on the elements of the first order transfer function vector, are therefore formed from this

---

\*This is assuming a single input circuit.

column of  $[Y(s)]^{-1}$ . The second-order transfer function vector is obtained by pre-multiplying the second-order current source vector by  $[Y(s_1+s_2)]^{-1}$ , according to which the second-order transfer function vector eventually depends on the entries of inverse of the node admittance matrix evaluated at  $(s_1+s_2)$ . The third- and higher-order transfer functions have a similar dependence. Thus, if the inverse of the reduced node admittance matrix is obtained in symbolic form, with  $s$  retained as a symbol, then a functional description of the nonlinear functions can also be obtained. A concomitant advantage of this functional description is that theorems from multi-dimensional theory [5] (such as initial value, final value, etc.) can then be applied to these transfer functions to gain more insight into the workings of the circuit.

In Chapter 2 we developed recursive relationships to estimate the error incurred in the truncation of the series solution. This error was directly related to the  $L_1$  norm of the linear kernel function, which, in turn, is related to the poles and residues of the linearized system. Thus, we can get an estimate of the accuracy of our solution through the pole-residue information provided to us by the symbolic analysis.

### 5-3. Symbolic Analysis Method

Symbolic circuit analysis by digital computer has been of considerable interest in the past decade. Many algorithms and methods have been derived to obtain symbolic transfer functions of linear circuits [29]. Most of these methods use tree enumeration [46], signal-flow graphs [29], or purely numerical methods [47] to obtain symbolic transfer function between the input and the output. These approaches are basically useful for single-input, single-output systems. The inversion of the

reduced node admittance matrix to obtain the open-circuit impedance matrix, which is the problem we are dealing with, is basically a multi-input, multi-output problem. The methods mentioned above can be adapted to solving the problem at hand; however, the generation of multiple symbolic functions using these approaches may not be satisfactory because of excessive computer time requirements. Some other approach is definitely warranted.

Published methods [25-27] for inverting the nodal admittance matrix when the elements are rational functions of the Laplace transform variable  $s$  use pivotal techniques. It may appear that, since it is easy to program a computer to perform polynomial arithmetic, these pivotal techniques are a natural way to approach the symbolic inversion problem. Results from the use of such a technique have proved to be disappointing, mainly due to the following reasons:

(a) The process of inversion transforms the nodal admittance matrix, which contains terms of the form  $a + \frac{b}{s} + c$ , into a matrix in which every element is a rational function of  $s$ . The pivotal technique produces the inverse matrix where common factors appear between numerator and denominator, and unless some mechanism is built into the process whereby these common factors are recognized and removed, the elements produced will have polynomials of excessively high order.

(b) When the circuit complexity is high, the evaluation of the symbolic function at high frequency values can give rise to numerical problems. For example, a circuit with 8 poles will have an  $s^8$  term in the characteristic polynomial. When evaluated at 10 Mrad/sec, this term produces a number equal to  $10^{56}$ . Of course, this problem can be alleviated by

obtaining a partial fraction expansion (PFE) form for the transfer functions. But this again entails additional computations - not to mention the numerical instability problems involved in root finding.

(c) It has also been found that pivotal techniques become numerically unstable for higher order circuits.

We therefore seek another alternative for obtaining the symbolic open circuit impedance matrix.

An approach based on the state variable formulation can be used to achieve this goal. Specifically, consider the general p-port augmented linear circuit of Fig. 5-1(a). We wish to solve for the transfer impedances,  $z_{ij}(s)$ ,  $i, j = 1, 2, \dots, p$ , from the j-th port to the i-th port. Knowing these transfer impedances, we can write for the p-port:

$$\underline{V}(s) = \underline{Z}(s)\underline{I}(s) = [\underline{Y}(s)]^{-1} \underline{I}(s) \quad (5-5)$$

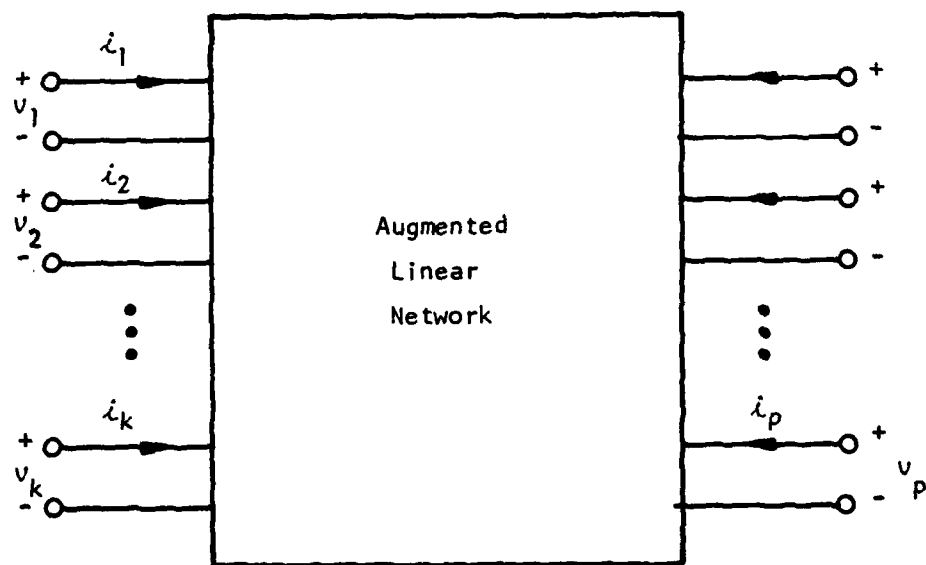
$$\text{where } \underline{V}(s) = [V_1(s) \ V_2(s) \ \dots \ V_p(s)] \quad (5-6)$$

$$\underline{Z}(s) = [z_{ij}(s)] \quad (5-7)$$

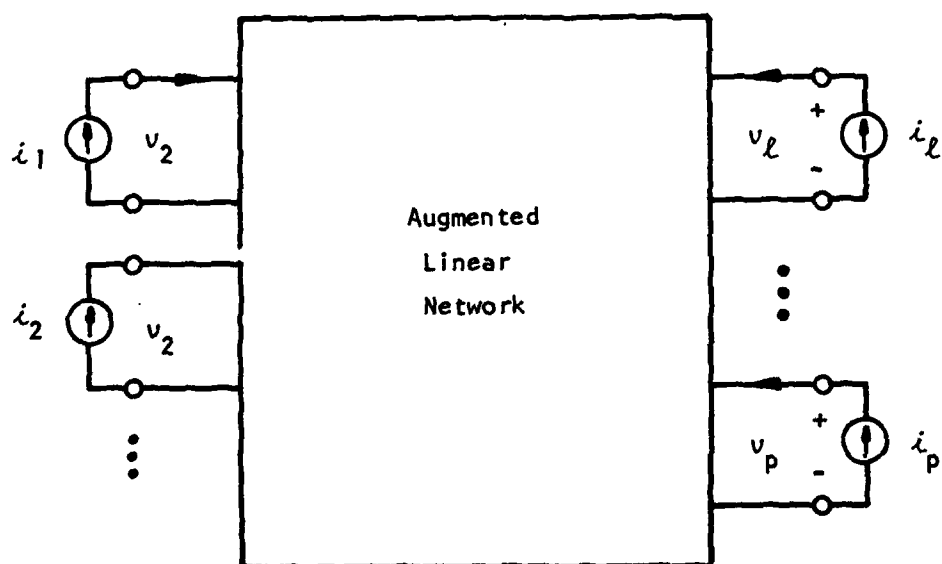
$$\text{and } \underline{I}(s) = [I_1(s) \ I_2(s) \ \dots \ I_p(s)] \quad (5-8)$$

Note that the vector  $\underline{V}(s)$  contains entries which are the output voltages and voltages that control the nonlinear element characteristics in the nonlinear circuit.

To obtain  $\underline{Z}(s)$  symbolically, we write for the network of Fig. 5-1(b), the following state equations:



(a)



(b)

Figure 5-1. Determination of  $[Y(s)]^{-1}=Z(s)$  for the  $p$ -port network using state equations.

$$\dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{i} \quad (5-9)$$

$$\underline{v} = \underline{C}\underline{x} + \underline{D}\underline{i} \quad (5-10)$$

where  $\underline{x}$  is the vector of state variables, and  $\underline{v}$  and  $\underline{i}$  are vectors whose transforms appear in eqns. (5-6) and (5-8), respectively. Taking the Laplace transform of eqn. (5-9) and (5-10), and solving for  $\underline{V}(s)$ , we get:

$$\underline{V}(s) = [\underline{C}(s\underline{I}-\underline{A})^{-1}\underline{B} + \underline{D}]\underline{I}(s) \quad (5-11)$$

and, therefore, we get  $\underline{Z}(s)$  to be

$$\underline{Z}(s) \equiv [\underline{C}(s\underline{I}-\underline{A})^{-1}\underline{B} + \underline{D}] \quad (5-12)$$

which is identically the inverse of the reduced node admittance matrix.

The matrix  $(s\underline{I}-\underline{A})$  can be inverted by applying the similarity transformation as follows:

$$\underline{A} = \underline{M}\underline{\Lambda}\underline{M}^{-1} \text{ or } \underline{\Lambda} = \underline{M}^{-1}\underline{A}\underline{M}$$

$$\therefore \underline{M}^{-1}(s\underline{I}-\underline{A})\underline{M} = s\underline{I} - \underline{M}^{-1}\underline{A}\underline{M} = s\underline{I} - \underline{\Lambda}$$

$$\text{or } (s\underline{I}-\underline{A})^{-1} = \underline{M}(s\underline{I}-\underline{\Lambda})^{-1}\underline{M}^{-1} \quad (5-13)$$

where the inverse of  $(s\underline{I} - \underline{\Lambda})$  is simply  $\text{diag } \{(s-\lambda_1)^{-1}, (s-\lambda_2)^{-1}, \dots\}$

where  $\lambda_i$  are the eigenvalues\* of the  $\underline{A}$  matrix and  $\underline{M}$  is the modal matrix.

Substituting eqn. (5-13) into eqn. (5-12), we get,

---

\*Here we assume distinct eigenvalues; the repeated eigenvalues can be handled similarly.

$$\begin{aligned} \underline{Z}(s) &= [\underline{C}\underline{M}(s\underline{I}-\underline{A})^{-1} \underline{M}^{-1} \underline{B} + \underline{D}] \\ &= [\hat{\underline{C}}(s\underline{I}-\underline{A})^{-1} \hat{\underline{B}} + \underline{D}] \end{aligned} \quad (5-14)$$

where  $\hat{\underline{C}} \triangleq \underline{C}\underline{M}$  and  $\hat{\underline{B}} \triangleq \underline{M}^{-1}\underline{B}$ . Equation (5-14) yields the entries of  $\underline{Z}(s)$  in partial fraction expansion form, which, as mentioned previously, is a more desirable form from a computational standpoint. All information regarding  $\underline{Z}(s)$  is contained in the matrices  $\hat{\underline{C}}$ ,  $\hat{\underline{B}}$ ,  $\underline{D}$  and a vector containing the eigenvalues. An algorithm for implementing this approach is given in Fig. 5-2. It should be noted that the approach used here is completely numerical and does not involve any coding and decoding of symbols.

Now that an algorithm for obtaining the symbolic  $\underline{Z}(s)$  is defined, we can make a comparison of the computational effort involved between using a symbolic inverse and the numerical inverse of the node admittance matrix at each frequency point.

The computational trade-off between the symbolic approach and a numerical approach for matrix inversion is very problem dependent. While a clear-cut winner cannot be established, a tentative answer can be obtained by noting the operations count, defined in terms of multiplications and additions, involved in the two schemes.

In the case of the numerical approach, the number of independent nodes,  $n$ , and the number of branches,  $b$ , are the most important quantities for determining the computational effort along with the number of frequency points at which the output is desired. Assuming that no sparse matrix techniques are used, the numerical inversion of an  $(n \times n)$  matrix requires  $O(n^3/3)$  units of work, where  $O(\ ) \equiv$  "order of", and 1



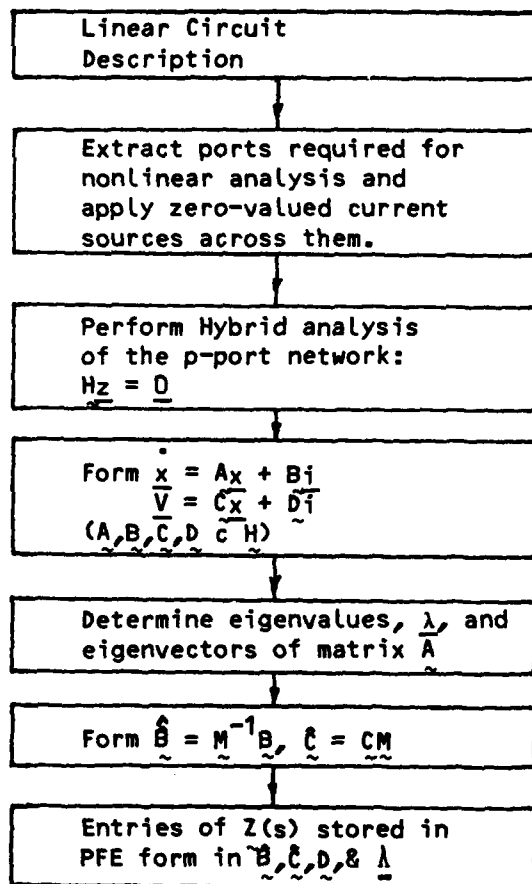


Fig. 5-2. Algorithm for inverting  $Y(s)$  symbolically.

unit of work = one addition and one multiplication. For  $k$  frequency points, the work becomes  $O(kn^3/3)$ . This does not involve book-keeping and other pre- and post-processing steps such as pivoting and iterative refinement, which are usually necessary to insure reliability and robustness of the algorithm.

In the case of symbolic inversion using our approach, the important parameters in the computational effort are the dynamic degrees of freedom,  $d$ , and the number of ports,  $p$ , where voltages and currents are injected or measured. Using the QR algorithm [29,48] for computing the eigenvalues of the  $A$  matrix, the operation count is  $O(8d^3)$ . The total work required for obtaining the inverse at  $k$  frequency points is therefore  $O(8d^3 + kdp^2)$ . The number,  $p$ , depends on the number of nonlinearities in the circuit, and is usually small. Also, if the network complexity is less than the number of nodes, the symbolic approach would, in general, require less computational effort. As far as accuracy is concerned, both the QR algorithm and the Crout's algorithm with pivoting and iterative refinement yield accurate results.

The efficiency of the symbolic method rests heavily upon the availability on an efficient process for forming the state equations. The hybrid analysis method [28,29], which essentially reduces to the analysis of a resistive network, is well-suited for our purposes here. We shall discuss this topic in section 5-6.

#### 5-4. Spectrum and Distortion Analysis Algorithm

The output spectrum and distortion indices for a nonlinear circuit with polynomial type nonlinearities can be computed on the basis of the material of Chapters 3 and 4. A flow-chart of the basic algorithm for

such a computation is given in Fig. 5-3. We describe the steps involved in the following paragraphs:

Step 1: For the given nonlinear circuit, determine the dc operating point. Expand each nonlinear function into a Taylor series about the operating point to get a polynomial representation for the nonlinear element in terms of the incremental quantities. Thus, for example, a forward-biased diode having the "global" V-I representation

$$I = I_s [\exp(qV/nkT) - 1] \quad (5-15)$$

can be expanded into a Taylor series to yield the following incremental v-i representation:

$$i = I_0 \frac{q}{nkT} v + \frac{I_0}{2!} \left(\frac{q}{nkT}\right)^2 v^2 + \frac{I_0}{3!} \left(\frac{q}{nkT}\right)^3 v^3 + \dots \quad (5-16)$$

where  $I_0$  is the dc operating current.

Step 2: Lump the linear part of the nonlinear elements with the existing linear network to form the augmented linear network. Extract as ports the nonlinear element branches and the branches that control the nonlinear element characteristics (dependent nonlinear element case), along with the output and source branches, from the augmented linear network. Let  $\underline{V} = [V_1 \ V_2 \ \dots \ V_p]$  and  $\underline{I} = [I_1 \ I_2 \ \dots \ I_p]$  denote the vector of voltages and currents for these ports, respectively.

Step 3: Using a symbolic analysis algorithm (see Fig. 5.2), obtain the entries of the Z matrix as a function of s, where

$$\underline{V}(s) = \underline{Z}(s) \underline{I}(s) \quad (5-17)$$

For each of the input sources, and their associated frequency tones,

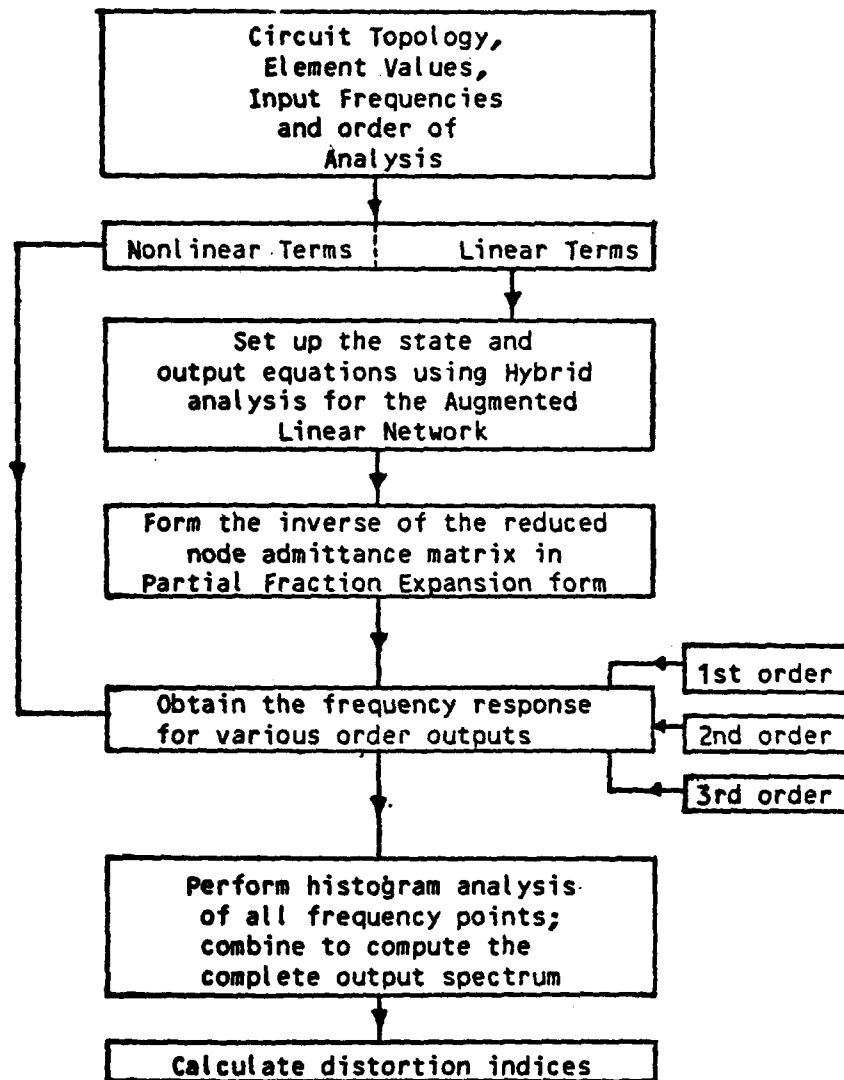


Figure 5-3. Algorithm for Spectrum and Distortion Analysis.

compute the first-order output voltages at each of the extracted ports by using the appropriate entries of the  $Z$  matrix. This step amounts to letting  $s = j\omega_i$  in  $z_{ij}(s)$ , the entries of  $\underline{Z}(s)$ .

Step 4: The second-order output spectrum is evaluated using the following relationship:

$$\underline{V}_2(s_1, s_2) = \underline{Z}(s_1 + s_2) \underline{I}_2(s_1, s_2) \quad (5-18)$$

The vector  $\underline{I}_2(s_1, s_2)$  is the second-order current source vector, which is formed by using the coefficients associated with the quadratic term of the nonlinear element and the first-order output at the controlling port(s) of the nonlinearity. The latter information was obtained in step 3. The given input tones are taken two at a time in eqn. (5-18), along with the information derived in section 4-2, to evaluate the output voltages at each of the p-ports.

The third-order output spectrum is obtained in exactly the same manner. The first- and second-order outputs are used to form the third-order current source at each combination frequency, which is then pre-multiplied by evaluating  $\underline{Z}(s)$  at the combination frequency.

Step 5: Perform a histogram analysis of all frequency points and combine the responses at points which are repeated. The distortion indices are computed using:

$$HD_2 = \frac{|V_0(2\omega_i)|}{|V_0(\omega_i)|} \quad (5-19)$$

$$HD_3 = \frac{|V_0(3\omega_i)|}{|V_0(\omega_i)|} \quad (5-20)$$

where  $HD_2$  and  $HD_3$  denote the second and third order harmonic distortion indices.

#### 5-5. Time-Domain Analysis Algorithm.

With  $Z(s)$  in eqn. (5-17) obtained in symbolic form, the zero-state time-domain response to Laplace transformable inputs can be determined by extending the algorithm discussed in section 5-4. This extension is carried on the basis of the material of section 4-3. Before outlining the new algorithm here, we derive relationships for obtaining the complete second- and third-order responses to a step input. The case of nonlinear capacitor and inductor is not treated here. As mentioned in [29], any nonlinear network can be equivalently represented by a nonlinear resistive network; we therefore treat this case here.

Consider a network with multiple nonlinear resistive elements, as shown in Fig. 5-4. As per steps 2 and 3 of section 5-4, we have the relationship (5-17) for the augmented linear network. Then, for a lumped linearized system, the transfer function  $H_1^i(s)$ ,  $i=1,2,\dots,p$ , is of the following form\*:

$$H_1^{(i)}(s) = \frac{V_i(s)}{V_{in}(s)} = z_{i1}(s) = \sum_{j=1}^N \frac{r_j^{(i1)}}{s+p_j} + \sum_{k=0}^M R_k^{(i)} s^k \quad (5-21)$$

Then, for a step input, the first-order output at each of these ports is:

---

\*Without any loss of generality, we have assumed that the source port is the port number 1.

$$Y_1^{(i)}(s) = \sum_{j=0}^N \frac{A_j^{(i)}}{s+p_j} + \sum_{k=0}^{M-1} q_k^{(i)} s^k \quad (5-22)$$

Notice that the first summation index starts from 0 instead of 1. This accounts for the  $A_0/s$  term appearing due to the step input.

The second-order transfer function can be expressed as (see eqn. 4-40 and 4-36):

$$H_2^{(i)}(s_1, s_2) = - \sum_{l=1}^L z_{il}(s_1+s_2) a_2^{(l)} H_1^{(l)}(s_1) H_1^{(l)}(s_2) \quad (5-23)$$

where  $L$  is the number of nonlinear elements,  $a_2^{(l)}$  is the coefficient of the quadratic term of the  $l$ -th nonlinearity. Then, the second-order output  $Y_2^{(i)}(s_1, s_2)$  is given by:

$$\begin{aligned} Y_2^{(i)}(s_1, s_2) &= H_2^{(i)}(s_1, s_2) X(s_1) X(s_2) \\ &= - \sum_{l=1}^L z_{il}(s_1+s_2) a_2^{(l)} H_1^{(l)}(s_1) X(s_1) H_1^{(l)}(s_2) X(s_2) \\ &= - \sum_{l=1}^L z_{il}(s_1+s_2) a_2^{(l)} Y_1^{(l)}(s_1) Y_1^{(l)}(s_2) \end{aligned} \quad (5-24)$$

Substituting eqn. (5-22) in eqn. (5-24) and removing any impulses, doublets, etc., we get the following:

$$Y_2^{(i)}(s_1, s_2) = - \sum_{l=1}^L z_{il}(s_1+s_2) a_2^{(l)} \left( \sum_{j=0}^N \frac{A_1^{(l)}}{s_1+p_j} \right).$$

$$\left( \sum_{k=0}^N \frac{A_k^{(L)}}{s_2 + p_j} \right) \quad (5-25)$$

$Y_2^{(i)}(s_1, s_2)$  can be reduced to  $Y_2^{(i)}(s)$ , the transform of  $y_2^{(i)}(t)$ , by using the association of variables [5]. This yields the following:

$$Y_2^{(i)}(s) = - \sum_{l=1}^L a_2^{(l)} \left[ \left( \sum_{d=1}^N \frac{r_j^{(il)}}{s + p_j} \right) \left( \sum_{n=0}^N \sum_{m=0}^N \frac{A_n^{(l)} A_m^{(l)}}{s + p_n + p_m} \right) \right] \quad (5-28)$$

Since the input is a step function, eqn. (5-28) can be simplified to\*:

$$Y_2^{(i)}(s) = \sum_k \frac{B_k^{(i)}}{s + p_k} + \sum_j \frac{C_k^{(i)}}{(s + p_j)^2} \quad (5-29)$$

where  $B_k$  and  $C_k$  are related to  $r_k$ 's and  $A_k$ 's, and  $p_k$  is related to the linear system poles,  $p_j$ . The second-order time response can be easily obtained from eqn. (5-29).

The third-order time domain response can be obtained similarly. We do not present it here in order to conserve space. Suffice it to say, the third-order response can be obtained from the values of the poles, and their associated residues, of the linearized system without performing any numerical integration or iteration. We now outline the algorithm for obtaining the zero-state time-domain response of a network with resistive nonlinearities (see Fig. 5-4):

**Step 1:** From the polynomial description of the resistive nonlinearities, form the augmented linear network. Extract as ports the nonlinear element branches along with the output and the source branches. Let

---

\*Here we have assumed that the linearized system had distinct poles. The case of multiple poles can be handled similarly, but it complicates the algorithm for computer implementation.



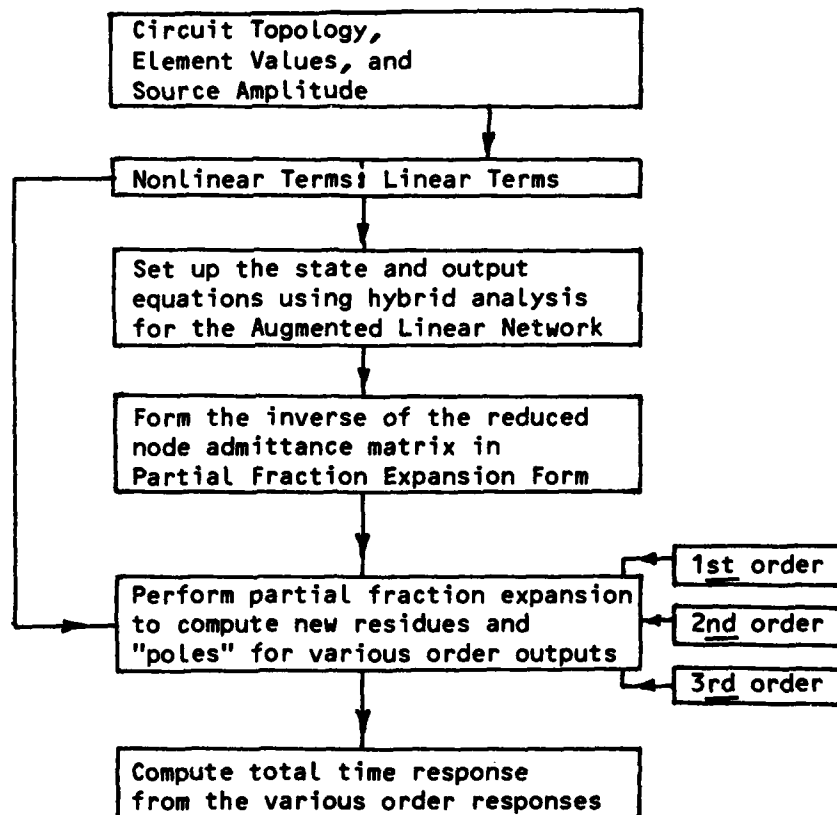


Fig. 5-4. Algorithm for computing time-domain response using Volterra series.

$\underline{V} = [V_1 \ V_2 \ \dots \ V_p]$  and  $\underline{I} = [I_1 \ I_2 \ \dots \ I_p]$  denote the vector of voltages and currents for these ports, respectively.

Step 2: Using a symbolic analysis algorithm (see Fig. 5-2), obtain the entries of the Z matrix in partial fractional expansion form, where

$$\underline{V}(S) = \underline{Z}(s) \underline{I}(S) \quad (5-30)$$

Step 3: From the pole-residue information obtained in step 2, obtain the first-order output response to the Laplace transformable input. This amounts to evaluating the sum of exponentials at each time step.

Step 4: Next obtain the second- and higher-order responses by appropriately associating the poles and residues of the linearized system. Thus, for example, in the second-order case, this amounts to determining  $B_k$ ,  $C_k$ , and  $p_k$  in equation (5-29). Once this information is obtained, the output is determined by evaluating terms of the form  $B_k \exp(p_k t)$  and  $C_k t \exp(p_k t)$  at each discrete time point.

#### 5-6. Program PRANC.

The Program for Analysing Nonlinear Circuits, known as PRANC, is a digital computer program, written in FORTRAN IV, that computes up to the third-order complete output spectrum of a nonlinear circuit with polynomial nonlinearities driven by up to two multi-frequency inputs.\* In the process it computes the Volterra transfer functions at each of the frequency combinations involved. PRANC also computes the zero-state time-domain response of circuits with only resistive nonlinear elements and a step input.

---

\*Thus, mixer-type circuits can be analyzed using PRANC.

As mentioned previously, the solution of the nonlinear circuit problem reduces to the repeated solution of the linear circuit. To efficiently handle this basic problem, PRANC uses a semi-symbolic approach [29] for analysing the augmented linear circuit. Specifically, the inverse of the reduced node admittance matrix is obtained in terms of the symbol  $s$  using the state equation formulation as described above.

The state equations for the linear circuit are formulated via the Hybrid analysis method [28,29]. If  $T$  denotes port branches in the tree [29] and  $C$  denotes port branches in the co-tree of a linear circuit, then the Hybrid analysis yields the following relationship:

$$\underbrace{\begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \\ H_{41} & H_{42} & H_{43} & H_{44} \end{bmatrix}}_{\underline{H}} \underbrace{\begin{bmatrix} i_T \\ v_C \\ v_T \\ i_C \end{bmatrix}}_{\underline{z}} = \underline{0} \quad (5-31)$$

By suitably forcing the various ports in the linear circuit into the tree and co-tree branches, PRANC uses the above formulation for setting up the state equations. All capacitor branches are extracted as ports which necessarily become part of the tree and all inductors, nonlinear element branches (which are assumed to be voltage controlled), and input and output branches, are extracted as ports which are forced as part of the co-tree. The matrix  $\underline{H}$  is obtained in a form where  $H_{11} = \underline{I}$  ( $\underline{I}$  being the identity matrix),  $H_{12} = H_{21} = 0$ ,  $H_{22} = \underline{I}$ . This yields the capacitor currents and the inductor and nonlinear element branch voltages in terms of known variables. Thus, the A, B, C, and D matrices in the state and

output equations (see eqns. 5-9 through 5-12) are obtained from the submatrices of  $\underline{H}$ . The formulation of eqn. (5-31) is quite fast, since it only involves the analysis of a resistive network.

It is noted that the matrix  $\underline{H}$  may not exist in idealized circuits. However, for most practical circuit this matrix is almost certain to exist [29]. It should also be noted that the above formulation of state equations tacitly assumes that no degenerate cutsets (all inductor-current source cutset) or degenerate loops (all capacitor-voltage source loop) are present in the linearized circuit. These restrictions are not very severe, especially when the realistic lossy models of circuit components are taken into account.

The next step in the PRANC algorithm is to determine the eigenvalues and the eigenvectors of the  $\underline{A}$  matrix. For this purpose, the double QR algorithm [48] for obtaining the eigenvalues is employed. The basic steps, such as matrix balancing, reduction to Hessenberg form, shift of origin, are employed in this algorithm to make it efficient and reliable. The eigenvectors are also obtained in the process.

All information about the inverse of the reduced node admittance matrix is stored as three matrices and a vector. The matrices are  $\hat{B}$ ,  $\hat{C}$ , and  $D$  (see eqns. 5-14), and the vector contains the eigenvalues. It is noted that the solution of eigenvectors for repeated eigenvalues can be a numerical unstable process [49]. Thus, the programs outputs a diagnostic message when such a case occurs.

The first-order voltage response at the prescribed ports is now computed from the entries of the open-circuit impedance matrix. These ports include: source port, output ports, nonlinear element ports, and

ports which control the nonlinear element characteristics. The response is calculated for each user prescribed frequency, and stored as a two-dimensional array: port number vs. the frequency number.

The second-order voltage response is computed at each distinct combination of the input tones taken two at a time. The ports of interest are the same as that for the first-order response. The second-order current source vector, at a particular frequency combination, is formed by considering the nonlinear element type and the voltage(s) controlling it, which is determined from the first order response array. This vector is pre-multiplied by the open-circuit impedance matrix evaluated at the combination frequency to obtain the second-order transfer function vector at that frequency. The response voltage at this frequency is then determined from the transfer function value. The second-order transfer function values are again stored as a two-dimensional array: port number and the particular frequency combination.

The third-order response is determined similarly. The third-order current source vector is formed by properly picking out the values of the first- and second-order transfer functions. The indexing of the arrays is of critical importance to the efficient implementation of this scheme.

When the time-domain response of a circuit with only nonlinear resistive elements is desired, the procedure is identical up until the formation of the symbolic open-circuit impedance matrix. The first-order step response is easily evaluated using the pole-residue information of the linearized circuit. The second-order response is evaluated by performing the association of poles and residues in a prescribed

manner. The third-order response is computed similarly. No numerical integration is used in these computations.

Since the hybrid analysis forms the basis for forming the open circuit impedance matrix, the following linear elements are allowed by the program\*: resistors, capacitors, inductors, voltage or current sources, and all four types of controlled sources. The nonlinear elements are assumed to be voltage controlled, with the following polynomial descriptions:

$$i_p = a_1 f[v_p] + a_2 f[v_p^2] + a_3 f[v_p^3] \quad (5-32)$$

$$i_p = a_{10} v_q + a_{01} v_r + a_{20} v_q^2 + a_{02} v_r^2 + a_{11} v_q v_r + a_{30} v_q^3 + a_{03} v_r^3 + a_{12} v_q v_r^2 + a_{21} v_q^2 v_r \quad (5-33)$$

where  $i_n$  and  $v_n$  are currents and voltages across branch  $n$ ,  $f$  is a linear operator of the type  $\frac{d}{dt}$ ,  $\int_{-\infty}^t$ , or constant, and  $a_{ij}$  are constants. It should be noted that eqn. (5-33) models a 3-port device.

In the present version, PRANC imposes the following restrictions on the circuit parameters: maximum number of elements (both linear and nonlinear) = 50; maximum number of nonlinear elements = 10; maximum number of dependent nonlinear elements (eqn. 5-33) = 5; maximum number of reactive elements = 20; maximum number of independent nodes = 30; number of input frequencies = 5. These restrictions can be relaxed if desired. The modular structure and algorithms of PRANC makes it possible to extend the order of analysis in a straightforward manner. The limit on

---

\*A direct nodal analysis would only allow for voltage controlled current source.

the highest order will eventually be dictated by the storage restrictions of the computer.

The validity of the results obtained from using PRANC has been verified through hand-worked examples and comparing with the results obtained from NCAP [44]. In the next chapter we will present two examples showing the results obtained from the use of PRANC.

## CHAPTER 6

### NONLINEAR CIRCUIT ANALYSIS EXAMPLES

#### 6-1. Introduction

In this section, results from the analysis of two nonlinear circuits using PRANC will be discussed. For the sake of brevity, the circuits are kept simple. A brief comparison of the execution times for PRANC and NCAP [44] is also included in this section.

Previous works [7,10,36] have used the Volterra series method to study the distortion phenomenon in transistor amplifiers. These authors show a good agreement between the predicted and the measured values of the response for small signal operation. Thus, the Volterra series is established as a viable approach for analyzing nonlinear circuits. In section 6-2, a single stage untuned amplifier is analyzed using PRANC.

Time-domain analysis using the Volterra series method has not received much attention in the literature. As such no practical circuits have been analyzed using this technique to study the validity of the results obtained. In section 6-3, the step response of a simple nonlinear circuit is obtained. These results are then compared with those obtained from a numerical integration method.

Finally, section 6-4 compares the execution times for the specific circuit examples analyzed using PRANC and NCAP.



## 6-2. Spectrum and Distortion Analysis Example.

Consider the single stage untuned bipolar transistor amplifier circuit shown in Fig. 6-1. This circuit configuration has been taken from [10]. The results obtained from using PRANC are in excellent agreement with those obtained from using NCAP, which, as reported in [10], are in good agreement with the measured results.

The equivalent circuit model [7] used for the bipolar transistor is shown in Fig. 6-2. It contains three nonlinear elements: a nonlinear resistor, a nonlinear capacitor, and a dependent nonlinearity. The operating point of the transistor is:  $V_{CE} = 10V$ ,  $I_C = 10mA$ . The pertinent parameters\* of the transistor are given in Table 6-1.

Table 6-1. Transistor Parameters for the Circuit of Fig. 6-1.

Parameter	Value	Parameter	Value
$I_E$	11.4mA	$n$	4.6
$I_C$	10.0mA	$a$	0.125
$I_{Cmax}$	150. mA	$r_b$	10.1 $\Omega$
$h_{FEmax}$	8.2	$r_e$	2.3 $\Omega$
$V_{CB}$	9.27 V	$r_c$	635k $\Omega$
$V_{CB0}$	140. V	$C_E$	1040pF
$u_{CB0}$	0.348	$C_{C1}$	11.1pF
$k$	25pF	$C_3$	1.5pF

The nonlinear elements, using the Taylor series expansion of the nonlinear functions, are represented by the following polynomials:

\*These parameters depend on the transistor type.

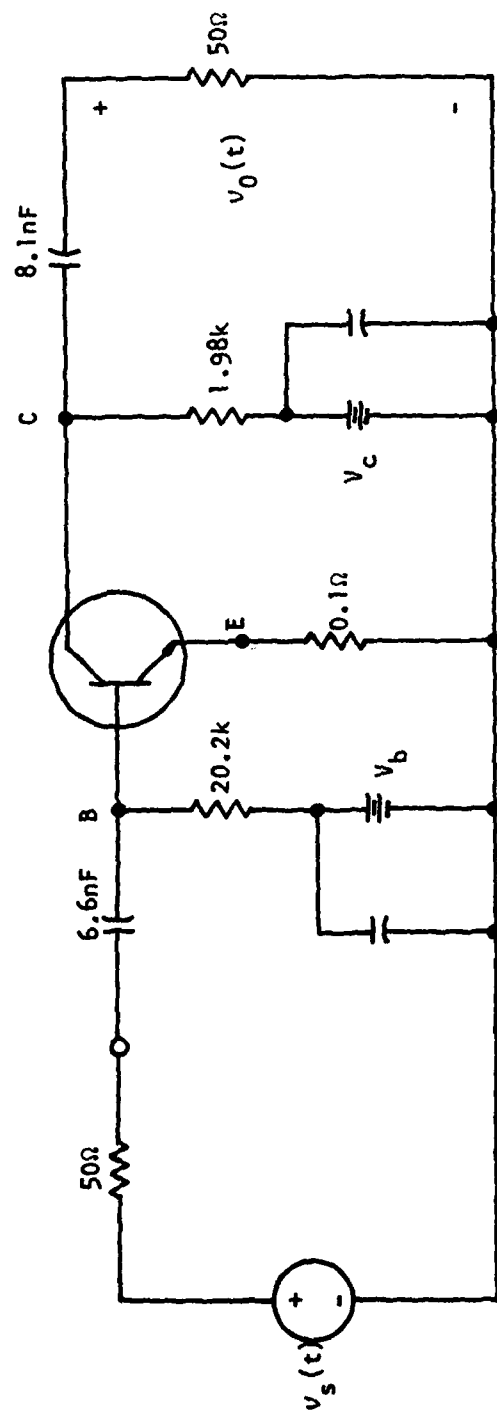


Figure 6-1. A Transistor Amplifier Circuit.

$$i_R = K(v_R) = g_1 v_R + g_2 v_R^2 + g_3 v_R^3 \quad (6-1)$$

$$i_C = P(v_C) = c_{C1} \frac{dv_C}{dt} + c_{C2} \frac{dv_C^2}{dt} + c_{C3} \frac{dv_C^3}{dt} \quad (6-2)$$

$$i_g = G(v_R, v_{cb}) = g_{10} v_R + g_{01} v_{cb} + g_{20} v_R^2 + g_{02} v_{cb}^2 \\ + g_{11} v_R v_{cb} + g_{30} v_R^3 + g_{03} v_{cb}^3 + g_{21} v_R^2 v_{cb} + g_{12} v_R v_{cb}^2 \quad (6-3)$$

Using the analytical formulae for  $g_{ij}$ ,  $c_{Cij}$ , and  $g_{ij}$  derived in [7] and the transistor parameters given in Table 6-1, the numerical values of the coefficients of the polynomials appearing in eqns. (6-1) through (6-3) are obtained. These values are given in Table 6-2.

The values in Table 6-2 along with the linear element values, the circuit topology, and the source information are used as the input information for PRANC. The effects of the frequency and the input amplitude on the output response are discussed below.

**6-2.1. Effect of Frequency:** The effect of frequency on the system response is directly related to the effect of frequency on the transfer functions. In the following paragraph we elaborate on this concept.

The relationship of the first-order steady state response and the first-order (linear) transfer function is well understood. For the circuit of Fig. 6-1, the first order transfer function magnitude characteristics is shown in Fig. 6-3. It should be apparent that the amplifier has a fairly broadband first order response.

The magnitude characteristics of the second-order transfer function as a function of frequency can be plotted similarly. However, in this

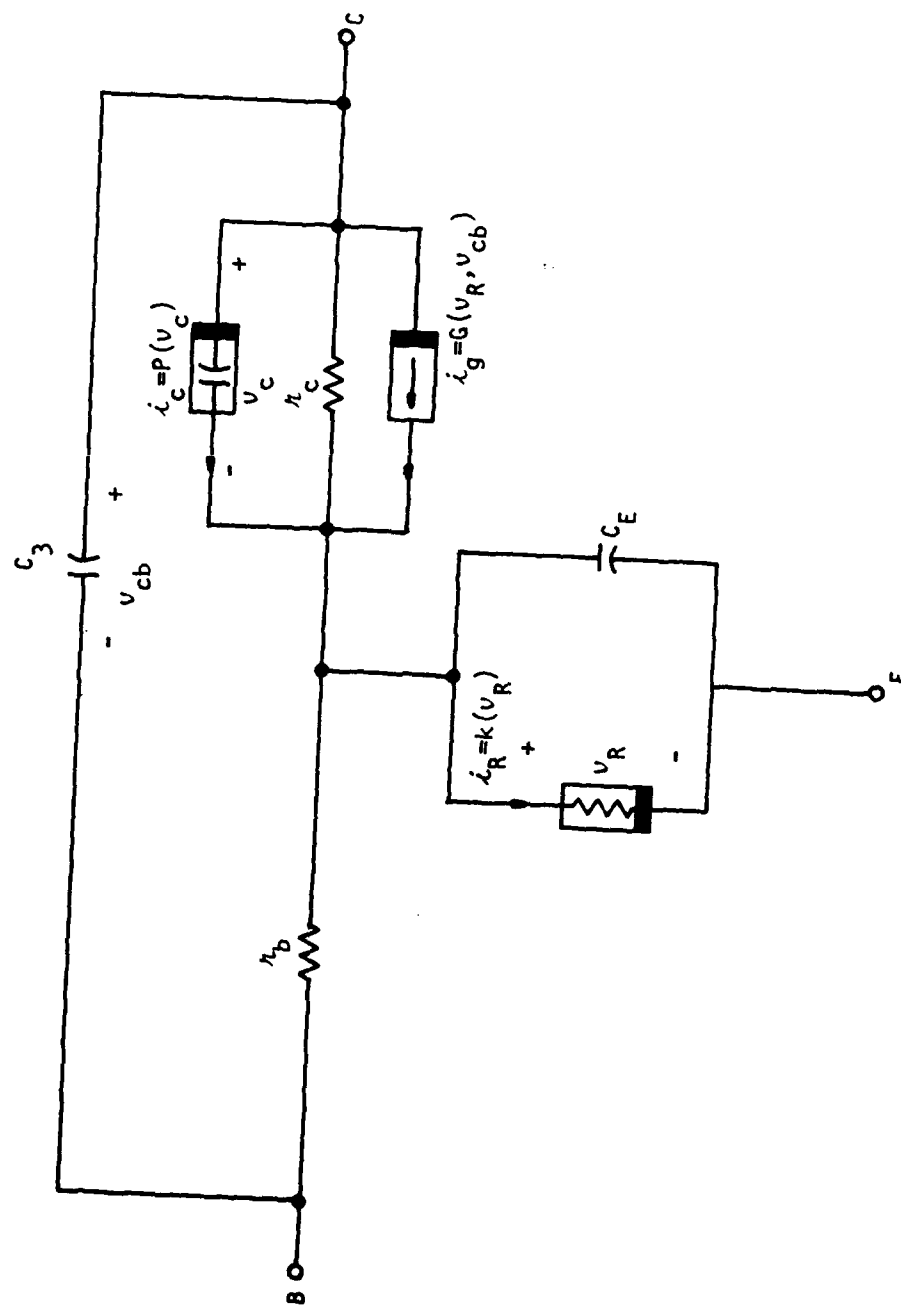


Figure 6-2. Transistor Equivalent Model.

Table 6-2. Coefficients of the polynomial type nonlinearities in Fig. 6-2.

Nonlinear Element	Coefficient Values	
Emitter Resistive Nonlinearity, $K(v_R)$	$g_1=0.4348$ $g_3=1.054 \times 10^{-2}$	$g_2=8.291$
Collector Capacitance Nonlinearity, $P(v_C)$	$c_{c1}=11.1 \times 10^{-12}$ $c_{c3}=4.784 \times 10^{-15}$	$c_{c2}=-1.996 \times 10^{-13}$
Collector dependent Nonlinearity, $G(v_R, v_{cb})$	$g_{10}=0.3856$ $g_{20}=7.419$ $g_{11}=7.215 \times 10^{-7}$ $g_{03}=3.397 \times 10^{-10}$ $g_{12}=1.401 \times 10^{-7}$	$g_{01}=1.871 \times 10^{-8}$ $g_{02}=3.633 \times 10^{-9}$ $g_{30}=94.66$ $g_{21}=1.388 \times 10^{-5}$

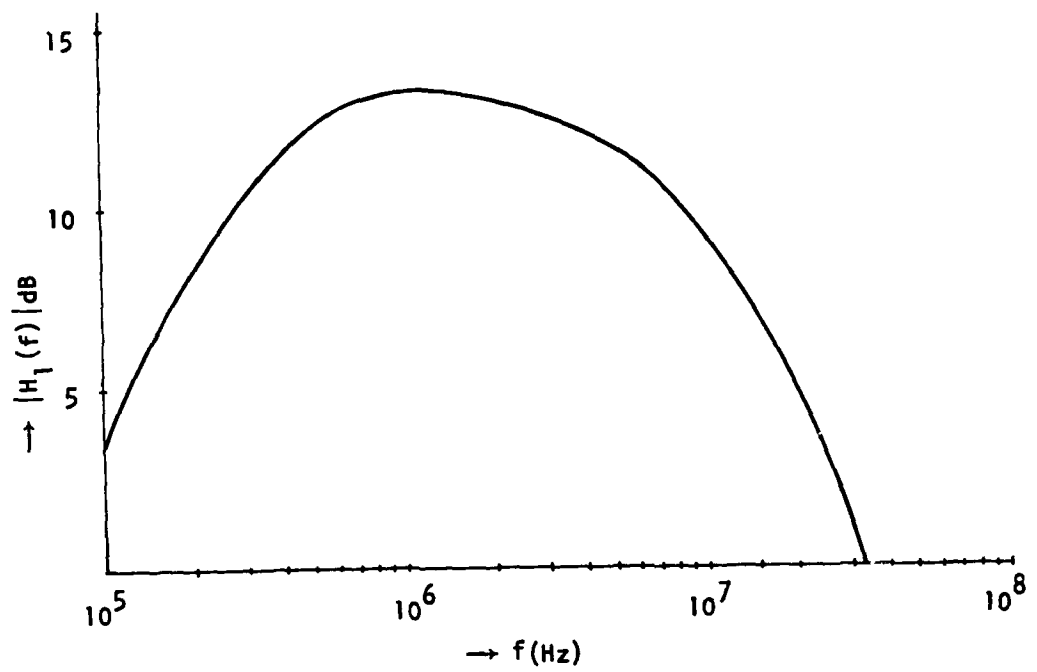


Figure 6-3. First-Order Transfer Function Magnitude Response.

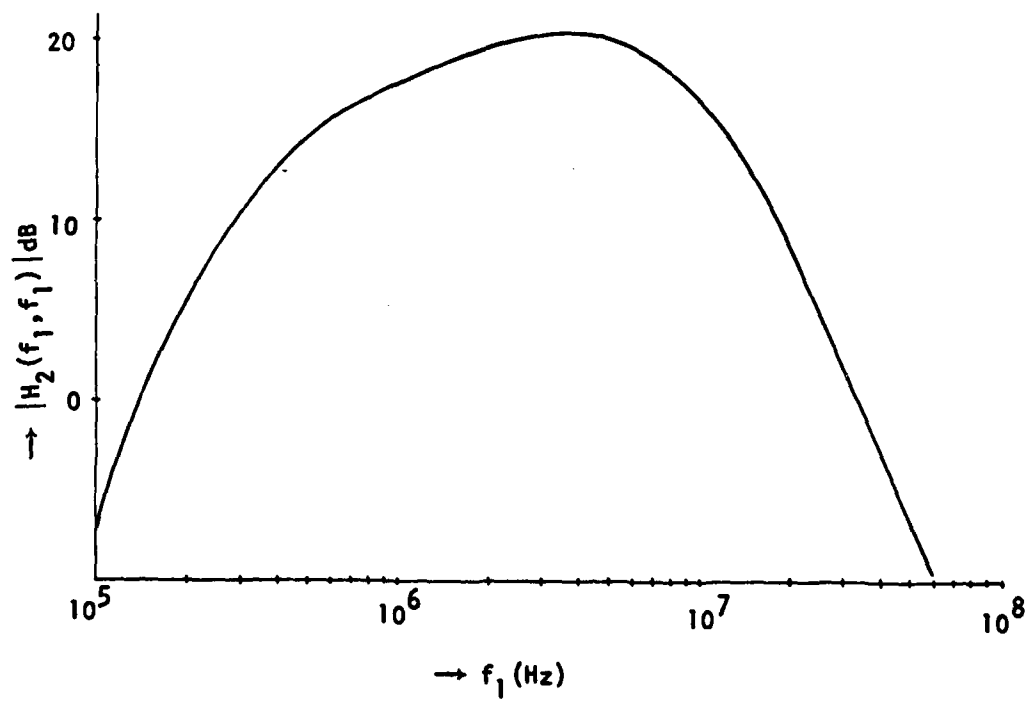


Figure 6-4. Second Harmonic Magnitude Response.

case, two frequencies are involved, and the significance of the magnitude plot must be clearly understood. An example of this plot is shown in Fig. 6-4, which depicts the magnitude characteristic of the second harmonic frequency response as the input tone is varied from 100KHz to 100MHz. Similarly, if the magnitude characteristics of an intermodulation frequency,  $\omega_{IM}$ , were desired, a similar plot of  $|H_2(j\omega_i, j\omega_k)|$ , such that  $\omega_i + \omega_k = \omega_{IM}$  in the band of interest, would provide the information.\* Thus, by letting  $\omega_i = |\omega_{IM} - \omega_k|$ , and sweeping  $\omega_k$  across the band of interest, the information on how the magnitude of an intermodulation product will vary can be obtained.

The magnitude characteristic of the third-order transfer function has a similar significance. In this case, a combination of three discrete frequencies is involved.

6-2.2. Effect of Input Amplitude. The effect of the input amplitude on the distortion products for the amplifier circuit is shown in Fig. 6-5. A single tone at 2.5MHz is used as the input to the amplifier and the second and third harmonic outputs are computed. On the double logarithmic paper, we note that the distortion plots are straight lines, with  $m_3 > m_2 > m_1$ , where  $m_1$ ,  $m_2$ , and  $m_3$  are the slopes of the first, second, and third harmonic distortion, respectively. It is noted that the validity of the small-signal model extends up to approximately 25mV input amplitude, after which the Volterra series prediction is more pessimistic than the measured results for the amplifier [10]. Also, the harmonic distortion indices for a given input amplitude can be deter-

---

\*Since PRANC computes the entire output spectrum, it automatically takes the negative frequencies into account. Thus the user does not have to specify, for example, a - 2.5MHz and 3MHz frequency to get the intermodulation product at 0.5MHz.

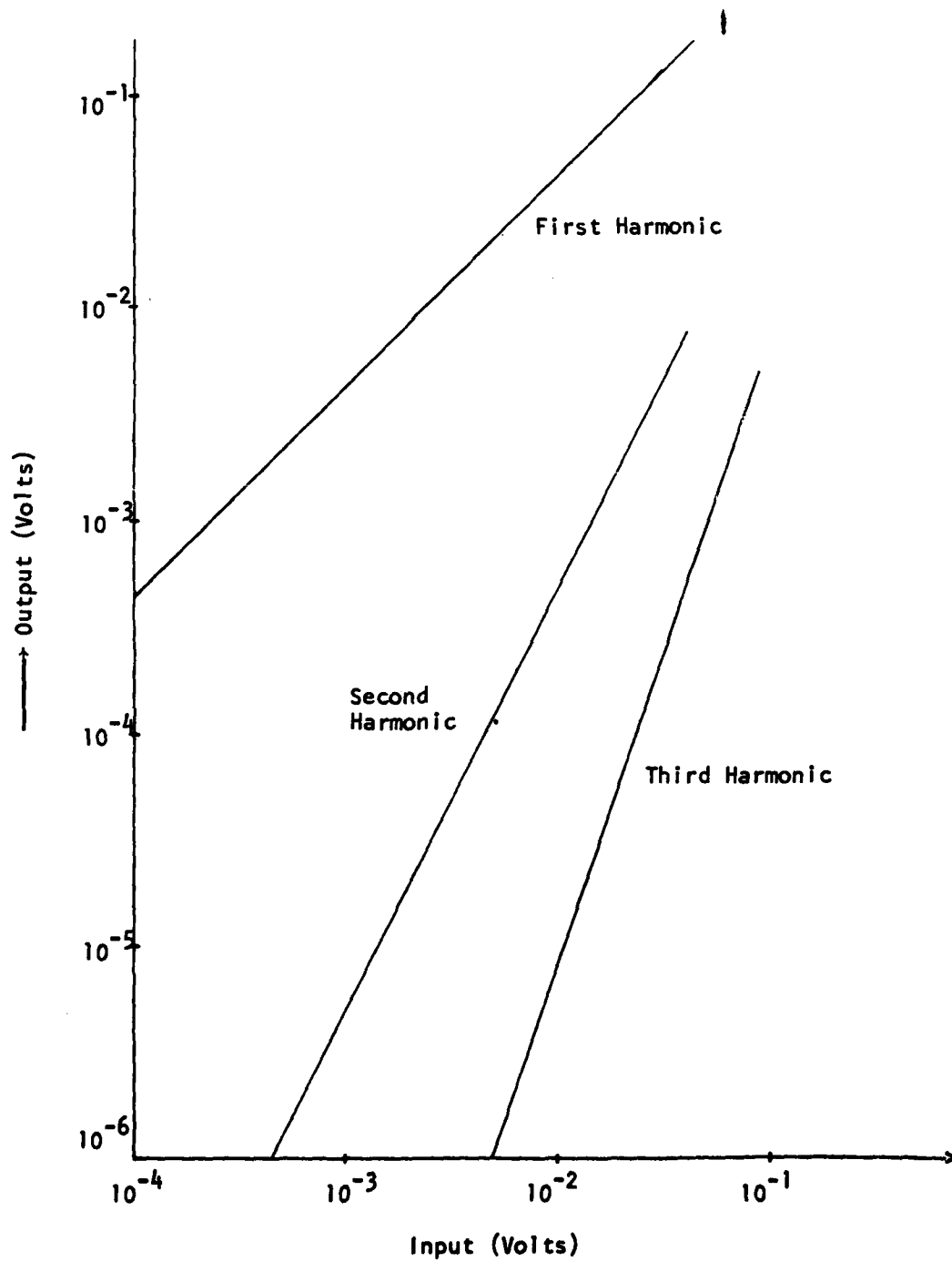


Figure 6-5. Harmonic Distortion Variation With Input Amplitude.



mined directly from Fig. 6-5.

6-3. Time-Domain Analysis Example. Consider the simple resistive non-linearity circuit of Fig. 6-6. The v-i characteristics for the non-linear element is  $i = 10v^3$ . In this section we will determine the step response of the circuit using Volterra series method and then compare the results obtained from an exact numerical integration method.

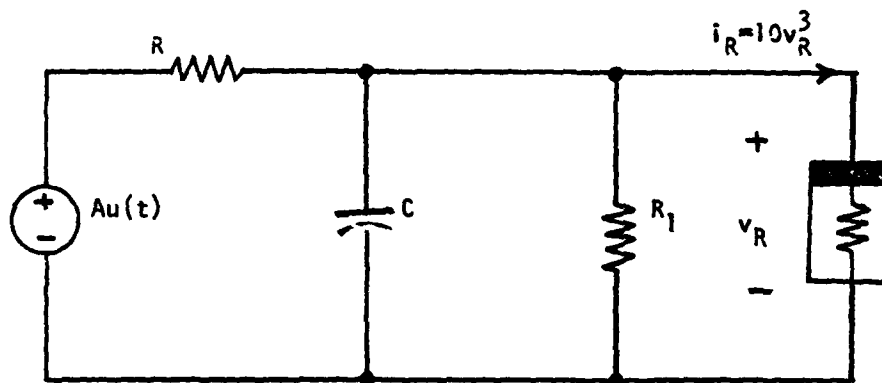


Fig. 6-6. A Nonlinear Circuit

By inspection, the first-order transfer for the circuit is:

$$H_1(s_1) = \frac{r}{s_1 + p} ; r = \frac{1}{RC}, p = \frac{G_1 + G}{C} \quad (G = \frac{1}{R}, G_1 = \frac{1}{R_1}) \quad (6-4)$$

For an input  $Au(t)$ , the first order response,  $y_1(t)$ , is given by

$$y_1(t) = a(1 - e^{-pt}) ; a = \frac{Ar}{p} \quad (6-5)$$

Since there is no square term in the nonlinearity, the second-order transfer function is zero. The third-order transfer function is found to be:

$$H_3(s_1, s_2, s_3) = -\frac{10}{C} \left( \frac{1}{(s_1 + s_2 + s_3) + (G + G_1)/C} \right) H_1(s_1) H_1(s_2) H_1(s_3) \quad (6-6)$$

and, therefore, the third-order output:

$$Y_3(s_1, s_2, s_3) = H_3(s_1, s_2, s_3) \frac{A^3}{s_1 s_2 s_3} \quad (6-7)$$

Substituting eqn. (6-6) in eqn. (6-7), and carrying out the association of variables, the third-order response is given by:

$$Y_3(s) = -K \left[ \frac{1}{s} - \frac{1.5}{s+p} - \frac{3p}{(s+p)^2} + \frac{0.5}{s+3p} \right], \quad K = 10 A^3/Cp \quad (6-8)$$

Therefore, the approximate response of the circuit of Fig. 6-8 is given by:

$$y(t) = y_1(t) + y_3(t)$$

$$= a(1 - e^{-pt}) - K[1 - 1.5\exp(-pt) - 3p\exp(-pt)$$

$$- 3\exp(-2pt) + 0.5\exp(-3pt)] \quad (6-9)$$

The response  $y(t)$  can be calculated using eqn. (6-9). The step response of the circuit of Fig. 6-8 is not as interesting as the error incurred in using Volterra series. A summary of the error vs. the  $l_1$  norm of the linear kernel function is given in Table 6-3. The norm is varied by changing the values of  $R_1$  in the circuit. This alters the value of  $p$ , but not of  $r$ , in eqn. (6-4). An estimate of the truncation error as predicted by Theorem 2-2 is also given in Table 6-3. Clearly, there is a good agreement between the calculated and the predicted values of this

error.

Table 6-3. Summary of Truncation Error

$L_1$ Norm of $h(\tau)$	Predicted Error* $O(30ZG^5X^5)$	Calculated Error	
		Maximum	Steady State
0.1333	$10^{-2}$	$10^{-2}$	$10^{-2}$
0.0909	$10^{-3}$	$10^{-3}$	$10^{-3}$
$9.9 \times 10^{-3}$	$10^{-9}$	$10^{-5}$	$10^{-10}$

\*"Order of "

#### 6-4. PRANC vs. NCAP

Both PRANC and NCAP [44] are based on the Volterra series method for analyzing nonlinear circuits. The basic algorithms used by the two programs are, however, different. PRANC uses a semi-symbolic analysis procedure, where as NCAP uses a purely numerical approach. Furthermore, the features offered by the two programs are different. NCAP allows a free-format input and computes the transfer functions and node voltages at  $2^n - 1$  discrete frequency points, where  $n$  is the order of the analysis.\* PRANC, on the other hand, requires a fixed format input, but computes the complete output spectrum. Thus, the number of output fre-

\*For NCAP, this happens to be the number of input frequencies also, since the user must specify  $n$  input tones in order to obtain an  $n$ -order analysis.

quency points for PRANC is much higher than that for NCAP. Because of the algorithm and the feature differences, the execution times required by the two programs for analyzing similar circuits are different. In the following paragraph we compare the approximate execution times for the two programs on the CDC 6500 computer at Purdue University.

The bipolar transistor amplifier of Figure 6-1 was analyzed on both PRANC and NCAP. For a two-tone input and up to a third-order analysis, PRANC required 1.74 sec to compute the complete output spectrum (involving approximately 32 frequency points). For a 5-tone input, it required 6 seconds to do the same job (approximately 150 points). By contrast, NCAP required 7.5 sec to compute the response at 7 discrete frequencies for a three tone input. Other examples exercised on PRANC and NCAP show a similar advantage in execution times: PRANC, despite providing more information to the user, requires less execution time than NCAP. A more detailed comparison of the two programs is contained elsewhere [41].

CHAPTER 7  
CONCLUDING REMARKS

7-1. Summary

The contents of this thesis has dealt with the various aspects of Volterra series analysis of lumped systems, with polynomial type nonlinear elements, with special emphasis on nonlinear circuits. The fundamental intent behind this work is to show that the Volterra series method could be used in a variety of engineering calculations - particularly computer-aided analysis of mildly nonlinear systems.

By investigating a concrete, yet general, set of differential equation that characterize lumped nonlinear systems, a unified theory has been developed here which can be used to study the behavior of such systems.

By analyzing how the series solution converges, simple and easy to use recursive relationships have been derived. These relationships can be used for: 1) estimating the error incurred in truncating the series; and 2) determining the bound on the input function for which the series converges. The higher-order kernel functions are not required a priori in these relationships; only the linearized system need be analyzed to determine how, if at all, the series converges.

The determination of the transfer function by the direct application of the multi-dimensional transforms has been developed here. This approach provides a more direct and rigorous way of characterizing the system using Volterra series.

The entire investigation culminated in the development of a computer program which, through the use of semi-symbolic analysis, provides an

efficient tool for spectrum and distortion analysis of mildly nonlinear circuits. The time-domain analysis of nonlinear circuits has also been investigated.

#### 7-2. Further Research

There are several problems that were unraveled during the course of this investigation. We mention a few of them here.

In dealing with the convergence of a class of lumped nonlinear systems, we showed that the truncation error involved three quantities: 1) the  $L_1$  norm of the linearized system, 2) the bound on the system input, and 3) the coefficients of the polynomial describing the nonlinear elements. An important consequence of this result leads to the following question: How do we speed up the convergence of the series solution and yet obtain meaningful results? One factor which has a direct bearing on the answer to this question is the criterion and method used for approximating the nonlinear function by polynomials. By changing the coefficient values, the convergence would also change. These coefficients depend on the norm of the error function we choose to minimize when approximating the nonlinear function by a polynomial. Thus, the need for establishing a criterion (choice of a norm) to arrive at a rapidly converging series is important.

Another problem is related to the interpretation of the nonlinear transfer functions. We have shown that these can be obtained directly from the application of multidimensional transforms. In the case of linear systems, the poles and zeros have a direct relation to the system response and stability. Since the Volterra series solution regards the linear system as a limiting case of the nonlinear system, an important

question to ask is: Is there a relation between the nonlinear transfer functions and the system stability? How does one interpret, for example, the significance of  $p_1$  in the  $(s_1 + s_2 + p_1)$  term in the second-order transfer function? If these questions can be adequately answered in terms of the circuit parameters, a significant insight into the behavior of nonlinear systems would be gained. As pointed out earlier, working in the transform domain is much simpler and algorithmic as compared to working in the time domain.

The synthesis problem using Volterra series has not received much attention. It is an important problem, one which can be used in a variety of applications. One of them, for example, could be to improve the received signal-to-noise ratio of a signal subjected to nonlinear processing.

#### REFERENCES

- [1] Volterra, V., Theory of Functionals and of Integral and Integro-Differential Equations, New York: Dover, 1959.
- [2] Wiener, N., Nonlinear Problems in Random Theory, New York: Technology Press, 1958.
- [3] Zames, Z. D., "Nonlinear Operators for System Analysis," MIT Research Lab. of Electronics, Tech. Rept. 370, Aug. 25, 1960.
- [4] Brilliant, M. B., "Theory of the Analysis of nonlinear systems," MIT Research Lab. of Electronics, Tech. Rept. 345, March 3, 1958.
- [5] George, D. A., "Continuous Nonlinear Systems," MIT Research Lab. of Electronics, Tech. Rept. 355, July 24, 1959.
- [6] Barrett, J. F., "The Use of Functionals in the Analysis of Nonlinear Physical Systems," Part-I, Sect. 2, A. T. Fuller, ed., Nonlinear Stochastic Control Systems, London: Taylor and Francis, 1970.
- [7] Narayanan, S., "Transistor Distortion Analysis Using Volterra Series Representation," BSTJ, 46, pp. 991-1023, May-June 1967.
- [8] Van Trees, H., Synthesis of Optimum Nonlinear Control, Mass.: MIT Press, 1962.
- [9] Van Trees, H., "Functional Techniques for the Analysis of the Nonlinear Behavior of Phase Locked Loops," Proc. IEEE, Vol. 52, No. 8, pp. 894-911, Aug. 1964.



- [10] Graham, J. and Ehrman, L., eds., Nonlinear System Modeling and Analysis with Application to Communication Receivers, Rome Air Development Center, RADC-TR-73-178 (AD-766278), Griffis AFB, June 1973.
- [11] Ewen, E., "Black Box Identification of Nonlinear Volterra Systems," TIS-R77EML7, General Electric Company, Aerospace Electric Systems Division, March 1977.
- [12] Bedrosian, E. and Rice, S., "The Output Properties of Volterra Systems (Nonlinear Systems with Memory) Driven by Harmonic and Gaussian Inputs," Proc. IEEE, Vol. 59, No. 12, pp. 1688-1707, Dec. 1971.
- [13] Naditch, G. H., A Scattering Variable Approach to Nonlinear Systems, Rome Air Development Center, RADC-TR-74-106, Griffis AFB, May 1974, 780535/IGI.
- [14] Bussgang, J. J., et. al., "Analysis of Nonlinear Systems with Multiple Inputs," Proc. IEEE, Vol. 62, No. 8, pp. 1088-1119, Aug. 1974.
- [15] Leon, B. J., Lumped Systems, NY: Hold, Rinehart and Winston, Inc., 1968.
- [16] Schaefer, D. J. and Leon, B. J., "The Steady State Analysis of Nonlinear Systems," Purdue University, TR-EE 76-2, Jan. 1976.
- [17] Hayashi, C., Nonlinear Oscillations in Physical Systems, McGraw-Hill, 1964.

- [18] Aggarwal, J. K., Notes on Nonlinear Systems, Van Nostrand Reinhold, 1972.
- [19] Leon, B. J., and Schaefer, D. J., "Volterra Series and Picard Iteration for Nonlinear Circuits and Systems," IEEE Trans., CAS-25, No. 9, pp. 789-793, Sept. 1978.
- [20] Gilbert, E. G., "Functional Expansions for the Response of Nonlinear Differential Systems," IEEE Trans. AC-22, No. 6, pp. 909-921, Dec. 1977.
- [21] Lesiah, C., and Krener, A. J., "The Existence and Uniqueness of Volterra Series for Nonlinear Systems," IEEE Trans., AC-23, No. 6, pp. 1090-1095, Dec. 1978.
- [22] Ku, Y. H., and Wolf, A. A., "Volterra-Wiener Functionals for the Analysis of Nonlinear Systems," Journal of The Franklin Institute, Vol. 281, No. 1, pp. 5-26, Jan. 1966.
- [23] Barrett, J. F., "The Use of Volterra Series to Find Region of Stability of a Nonlinear Differential Equation," International Journal of Control, Vol. 1, pp. 209-216, 1965.
- [24] Lubbock, J. K., and Bansal, V. S., "Multidimensional Laplace Transforms for Solution of Nonlinear Equations," in Multidimensional Systems: Theory and Applications, N. K. Bose, ed., IEEE Press, pp. 155-162, 1979.
- [25] Down, T., "Symbolic Evaluation of Transmittances from the Nodal Admittance Matrix," Elec. Letters, Vol. 5, No. 16, pp. 379-380, Aug.

1969.

- [26] -, "Inversion of the Nodal-Admittance Matrix in Symbolic Form,"  
Elect. Letters, Vol. 6, No. 3, pp. 74-76, Feb. 1970.
- [27] -, "Inversion of the Nodal Admittance Matrix for Active Networks in  
Symbolic Form," Elect. Letters, Vol. 6, No. 22, pp. 690-691, Oct.  
1970.
- [28] Lin, P. M., "Formulation of Hybrid Matrices for Linear Multiports  
Containing Controlled Sources," IEEE Trans. Vol. CT-21, pp.  
169-175, March 1974.
- [29] Chua, L. O., and Lin, P. M., Computer-Aided Analysis of Electronic  
Circuits: Algorithms and Computational Techniques, Englewood  
Cliffs, NJ: Prentice-Hall, 1975.
- [30] Leon, B. J., and Anderson, D. R., "The Stability of Pumped Non-  
linear Reactance Circuits," IEEE Trans., Vol. CT-10, No. 4, pp.  
468-476, Dec. 1963.
- [31] Trick, T. N. and Aprille, T. J., "Steady State Analysis of Non-  
linear Circuits with Periodic Inputs," Proceedings IEEE, Vol. 60,  
No. 1, Jan. 1972.
- [32] DeCarlo, R. A., Thapar, H. K., and Stuffle, R. E., "A Steady State  
Analysis and Design Package for Interconnected Systems," Purdue  
Univ., School of EE, TR-EE 79-17, April 1979.

- [33] Dahlquist, G., Bjork, A., Numerical Methods, NJ: Prentice Hall, 1974.
- [34] Geyer, E. M., "Some Aspects of Volterra Series," Ph.D. Thesis, Purdue University, Aug. 1976.
- [35] Tricomi, F. G., Integral Equations, Interscience Publishers, Inc., New York, 1957.
- [36] Gopal, K., et. al., "Distortion Analysis of a Transistor Networks," IEEE Trans. Vol. CAS-25, No. 2, pp. 99-106, Feb. 1978.
- [37] Rudko, M. M. and Wiener, D. D., "Volterra Systems with Random Inputs: A Formalized Approach," IEEE Trans. Vol. COM-26, pp. 217-227, Feb. 1978.
- [38] Baranyi, A., "Nonlinear FM Distortion Equalizer," IEEE Trans., Vol. COM-26, No. 2, pp. 227-235, Feb. 1978.
- [39] Bansal, V. S., "Volterra Series Analysis of a Class of Nonlinear Time-Varying Systems Using Multilinear parametric transfer functions," Proc. IEE, Vol. 116, Nov. 1969, pp. 1957-1960.
- [40] Feller, W., An Introduction to Probability Theory and Its Applications, John Wiley and Sons, NY, 1957.
- [41] Thapar, H. K., and Leon, B. J., "PRANC: A Program for Analyzing Mildly Nonlinear Circuits," To be published as a Purdue University Tech. Report, Aug. 1979.

- [42] Thapar, H. K. and Leon, B. J., "Computer-Aided Distortion and Spectrum Analysis of Nonlinear Circuits," IEEE Int'l. Symposium on CAS, Tokyo, July 1979.
- [43] Kuo, Y. L., "Frequency-Domain Analysis of Weakly Nonlinear Networks," IEEE CAS Newsletter, CAS-11, Aug. and Oct. 1977.
- [44] "NCAP: Nonlinear Circuit Analysis Program," Rome Air Development Center, Griffis AFB, NYC, Private Communication.
- [45] Lin, P. M., "A Survey of Applications of Symbolic Network Functions," IEEE Trans., Vol. CT-20, pp. 732-737, Nov. 1973.
- [46] McClenahan, J., and Chan, S. P., "Computer Analysis of General Linear Networks Using Digraphs," Internl. J. Electronics, No. 22, pp. 153-191, 1972.
- [47] Singhal, K., and Vlach, J., "Symbolic Analysis of Analog and Digital Circuits," IEEE Trans., Vol. CAS-24, No. 11, pp. 598-609, Nov. 1977.
- [48] Parlett, B. N., "The LU and QR Algorithms," in A. Ralston and H. S. Wilf, eds., Numerical Methods for Digital Computers, NYC: John Wiley and Sons, Inc., 1967, Vol. II, pp. 116-130.
- [49] Acton, F. S., Numerical Methods that Work, Harper & Row, Publishers, NYC, 1970.

APPENDIX

A. For Chapter 2

A-1. Derivation of Eqn. 2-23:

According to eqn. (2-4):

$$y_2(t) = \int_0^{\infty} \int_0^{\infty} h_2(\tau_1, \tau_2) x(t-\tau_1) x(t-\tau_2) d\tau_1 d\tau_2 \quad (A-1)$$

From eqn. (2-19), we have

$$y_2(t) = -ca_2 \int_0^{\infty} \int_0^{\infty} h(t-\tau) R(\tau-\tau_1) R(\tau-\tau_2) \cdot x(\tau_1) x(\tau_2) d\tau_1 d\tau_2 d\tau \quad (A-2)$$

We make the following change of variables:

$$\begin{aligned} u &= t - \tau \implies \tau = t - u \\ u_1 &= t - \tau_1 \implies \tau_1 = t - u_1 \\ u_2 &= t - \tau_2 \implies \tau_2 = t - u_2 \end{aligned} \quad (A-3)$$

Substituting eqn. (A-3) in eqn. (A-2), we get

$$y_2(t) = -ca_2 \int_0^{\infty} \int_0^{\infty} h(u) R(u_1 - u) R(u_2 - u) \cdot x(t-u_1) x(t-u_2) du du_1 du_2 \quad (A-4)$$

Re-defining  $u = \tau$ ,  $u_1 = \tau_1$ ,  $u_2 = \tau_2$  in eqn. (A-4), we get:

$$y_2(t) = -\epsilon a_2 \int_0^\infty \int_0^\infty h(\tau) R(\tau_1 - \tau) R(\tau_2 - \tau) \cdot$$

$$x(t - \tau_1) x(t - \tau_2) d\tau d\tau_1 d\tau_2 \quad (A-5)$$

Comparing (A-5) with (A-1), we get

$$h_2(\tau_1, \tau_2) = \epsilon a_2 \int_0^\infty h(\tau) R(\tau_1 - \tau) R(\tau_2 - \tau) d\tau \quad (A-6)$$

#### A-2. Proof of Theorem 2-2:

Given

$$L(p) y(t) + \sum_{n=2}^{\infty} a_n y^n(t) = x(t)$$

Applying the inverse operator we get

$$y(t) + \int_0^\infty h(t - \tau) \left[ \sum_{n=2}^{\infty} a_n y^n(\tau) \right] d\tau = \int_0^\infty h(t - \tau) x(\tau) d\tau \quad (A-7)$$

Assuming a series solution for  $y(t)$ , such that

$$y(t) = y_1(t) + y_2(t) + y_3(t) + \dots \quad (A-8)$$

and substituting in eqn. (A-7), we get

$$\sum_{i=1}^{\infty} y_i(t) + \int_0^\infty h(t - \tau) \left[ \sum_{n=2}^{\infty} a_n \left( \sum_{i=1}^{\infty} y_i(\tau) \right)^n \right] d\tau = \int_0^\infty h(t - \tau) x(\tau) d\tau \quad (A-9)$$

Solving for  $y_1(t)$ ,  $y_2(t)$ , ... successively in eqn. (A-9), we get the following set of equations:

$$y_1(t) = \int_0^{\infty} h(t - \tau) x(\tau) d\tau$$

$$y_2(t) + a_2 \int_0^{\infty} h(t - \tau) y_1^2(\tau) d\tau = 0$$

$$y_3(t) + a_2 \int_0^{\infty} h(t - \tau) \left[ a_2 y_1(\tau) y_2^2(\tau) + a_3 y_1^3(\tau) \right] d\tau = 0$$

$$y_4(t) + \int_0^{\infty} h(t - \tau) i_n(\tau) d\tau = 0 \quad (\text{A-10})$$

where  $i_n(\tau)$  is shown [10,12] to be:

$$i_n(t) = \sum_{k=2}^n a_k y_{n,k}, \quad n \geq 2$$

$$y_{l,m} = \sum_{j=1}^{l-m+1} y_j y_{l-j,m-1} ; \quad y_{l,1} = y_l \quad (\text{A-11})$$

Then, for  $G = \int_0^{\infty} |h(\tau)| d\tau$  and

$$X = \max_{0 \leq t < \infty} [x(t)]$$

we get

$$y_1 = |y(t)| \leq GX$$

$$y_2 \leq |a_2| G^3 X^2$$

$$y_n \leq G |I_n|$$

where  $y_n$  is given by eqn. (2-38) after we use eqn. (A-11). The inequality (2-37) then follows from eqn. (A-8).



B. For Chapter 3

B-1. Derivation of  $\mathcal{L}[y^2(t)]$ :

Using the functional property of the  $\delta$ -function, we can write

$$y_1^2(t) = \int_{-\infty}^{\infty} \int y(\tau_1) y(\tau_2) \delta(t - \tau_1) \delta(t - \tau_2) d\tau_1 d\tau_2 \quad (\text{B-1})$$

Introducing dummy variables  $t_1$  and  $t_2$  in eqn. (B-1), we get

$$y(t_1, t_2) = y_1^2(t) = \int_{-\infty}^{\infty} \int y(\tau_1) y(\tau_2) \delta(t_1 - \tau_1) \delta(t_2 - \tau_2) d\tau_1 d\tau_2 \quad (\text{B-2})$$

Taking a two-dimensional transform of eqn. (B-2), as per eqns. (3-11) and (3-12), we get

$$Y(s_1, s_2) = \mathcal{L}[y(t_1, t_2)] = Y(s_1) Y(s_2) \quad (\text{B-3})$$

The derivation for  $\mathcal{L}[y^n(t)]$  is done similarly by introducing  $n$ -delta functions first, etc.

B-2. Derivation of  $\mathcal{L}\left[\frac{d}{dt} y^2(t)\right]$

Using the chain rule, we get

$$\frac{d}{dt} y^2(t) = 2y(t) \frac{d}{dt} y(t) = 2y(t) y'(t) \quad (\text{B-4})$$

Using eqn. (B-2), we re-write eqn. (B-4) as:

$$\begin{aligned} 2y(t) y'(t) &= \frac{1}{2} \left[ 2 \int_0^{\infty} \int y(\tau_1) y'(\tau_2) \delta(t_1 - \tau_1) \delta(t_2 - \tau_2) d\tau_1 d\tau_2 \right. \\ &\quad \left. + 2 \int_0^{\infty} \int y(\tau_2) y'(\tau_1) \delta(t_1 - \tau_1) \delta(t_2 - \tau_2) d\tau_1 d\tau_2 \right] \quad (\text{B-5}) \end{aligned}$$

Taking a two dimensional transform of eqn. (B-5), we get

$$Y(s_1, s_2) = \frac{1}{2} \left[ 2s_2 Y(s_1) Y(s_2) + 2s_1 Y(s_1) Y(s_2) \right] \quad (B-6)$$

$$= (s_1 + s_2) Y(s_1) Y(s_2)$$

The derivation of  $\mathcal{L} \left[ \frac{d}{dt} y^n(t) \right]$  is done similarly.

